

---

# Towards a Decentralized Publication System: A Proposal Using Blockchain and P2P Technologies

---



Master's Thesis

Viktor Jacynycz García

*Directed by*

Samer Hassan and Antonio Sánchez Ruiz-Granados

Máster en ingeniería informática

Master's degree in Computer Science Engineering

Facultad de Informática

Universidad Complutense de Madrid

2017/2018 academic year

Document made with T<sub>E</sub>X<sub>S</sub> v.1.1+. modified by Viktor Jacynycz García

This document is prepared for duple-side printing.

# Towards a Decentralized Publication System: A Proposal Using Blockchain and P2P Technologies

*Memoria que presenta para optar al master en ingeniería informática*

**Viktor Jacynycz García**

Calificación: **Matrícula de Honor**

*Directed by*

**Samer Hassan and Antonio Sánchez Ruiz-Granados**

**Máster en ingeniería informática**

**Master's degree in Computer Science Engineering**

**Facultad de Informática**

**Universidad Complutense de Madrid**

**2017/2018 academic year**

## Attribution-NonCommercial-ShareAlike 4.0 International

*CC BY-NC-SA 4.0*

You are free to:

- **Share** — copy and redistribute the material in any medium or format
- **Adapt** — remix, transform, and build upon the material

The licensor cannot revoke these freedoms as long as you follow the following terms:

- **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **NonCommercial** — You may not use the material for commercial purposes.
- **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
- **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.



<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Viktor Jacynycz García



El abajo firmante, matriculado en el Máster en Ingeniería Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: *Towards a Decentralized Publication System: A Proposal Using Blockchain and P2P Technologies*, realizado durante el curso académico 2017-2018 bajo la dirección de Samer Hassan y Antonio Sanchez Ruiz-Granados en el Departamento de ISIA, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en internet y garantizar su preservación y acceso a largo plazo.

Fdo. Viktor Jacynycz García



*A mi madre, por ser una mujer con super poderes capaz de afrontar cualquier  
dificultad, siempre has estado ahí*

*A mi padre, por ayudarme a ser como soy ahora, nunca me rendiré contigo*

*A mi hermano, por ser el mejor, haces que quiera seguir superándome para  
alcanzarte*

*A Jenny, por cambiarme la vida, siempre tiras de mí cuando lo necesito*



# Agradecimientos

*I find your lack of faith disturbing*  
Darth Vader, Star Wars: A New Hope.

Me gustaría agradecer a los profesores Samer Hassan y Antonio Sanchez Ruiz-Granados por el apoyo prestado para realizar este trabajo.

A Antonio Tenorio-Fornés y a David Llop por toda la ayuda que han prestado al diseño de este proyecto.

Al departamento de Ingeniería del Software e Inteligencia Artificial por fomentar mis ideas y potenciarlas.

A la Universidad Complutense por brindarme el conocimiento necesario para poder realizar esta plataforma y por ofrecerme un puesto en la Facultad de Informática.

A todos los que piensan que este trabajo no llegará a ningún lado, que me dan fuerzas para seguir adelante.



# Abstract

*I invent, transform, create and destroy for a living and  
when I don't like something about the World I change it.*

Rick and Morty - Dan Harmon, Justin Roiland

Science publication and peer review raises concerns about fairness, quality, performance, cost or accuracy. The Open Access movements has been unable to fulfill all its promises, and middlemen publishers can still impose policies and concentrate profits. This work, using emerging distributed technologies such as Blockchain and IPFS, proposes a decentralized publication system for open science called Decentralized Science<sup>1</sup>. It provides transparent governance, a distributed reviewer reputation system, and open access by-design.

**Keywords:** Peer review, reputation network, open access, blockchain, Ethereum, IPFS.

---

<sup>1</sup>Available at <https://decentralized.science>





# Resumen

*We do what we must because we can.*

GLaDOS - Portal.

El proceso de publicación científica y la revisión por pares generan inquietudes sobre la equidad, calidad, rendimiento, coste o precisión de este. Los movimientos del *Open Access* no han podido cumplir todas sus promesas, ya que las editoriales, que actúan como intermediarios, todavía pueden imponer políticas y concentrar gran parte de los beneficios económicos de este sistema. Este trabajo, utilizando tecnologías distribuidas emergentes como Ethereum o IPFS, propone un sistema descentralizado de publicación científica para la *ciencia libre* llamado Decentralized Science<sup>2</sup>. Proporciona un sistema de gobernanza transparente, un sistema de reputación de revisores distribuido y un diseño totalmente *Open Access*.

**Palabras Clave:** Revisión por pares, red de reputación, open access, blockchain, Ethereum, IPFS.

---

<sup>2</sup>Disponible en <https://decentralized.science>



# Contents

<b>Agradecimientos</b>	<b>ix</b>
<b>Abstract</b>	<b>xi</b>
<b>Resumen</b>	<b>xiii</b>
<b>I Introduction and previous considerations</b>	<b>1</b>
<b>1 Introduction, objectives and document structure</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Objectives . . . . .	5
1.3 Document Structure . . . . .	6
1.4 Introducción . . . . .	9
1.5 Objetivos . . . . .	11
1.6 Estructura del documento . . . . .	12
<b>2 Project background</b>	<b>13</b>
2.1 Socio-cultural background . . . . .	13
2.1.1 Publication systems . . . . .	13
2.1.2 Alternative publication systems . . . . .	15
2.1.3 Reputation systems . . . . .	17
2.2 Technical background . . . . .	19
2.2.1 Network Architectures . . . . .	19
2.2.2 Digital Currencies and Cryptocurrencies . . . . .	20
2.2.3 Bitcoin . . . . .	21
2.2.4 Ethereum . . . . .	23

2.3	Peer review and publishing software . . . . .	27
<b>3</b>	<b>Methodology &amp; Technology</b>	<b>31</b>
3.1	Methodology . . . . .	31
3.1.1	Project Timeline . . . . .	31
3.1.2	Brainstorming . . . . .	32
3.1.3	Value proposition canvas . . . . .	33
3.1.4	Agile methodologies . . . . .	35
3.2	Technology . . . . .	36
3.2.1	IPFS . . . . .	36
3.2.2	Ethereum . . . . .	39
3.2.3	Remix . . . . .	40
3.2.4	Testrpc . . . . .	41
3.2.5	JavaScript and Metamask . . . . .	42
3.2.6	Github . . . . .	43
<b>II</b>	<b>Decentralized Science</b>	<b>45</b>
<b>4</b>	<b>Platform description</b>	<b>47</b>
4.1	Design using a Decentralized Infrastructure . . . . .	47
4.2	Platform's features . . . . .	50
4.2.1	Distributed and Open Access By-Design . . . . .	50
4.2.2	A Distributed Reviewer Reputation System . . . . .	51
4.2.3	Transparent Governance . . . . .	52
<b>5</b>	<b>Architecture</b>	<b>53</b>
5.1	First Prototype . . . . .	53
5.2	Second prototype . . . . .	59
5.3	Ethereum addresses in Decentralized Science . . . . .	62
5.4	Reputation system . . . . .	62
5.5	Platform interactions and activity diagrams . . . . .	64
5.6	Transaction costs . . . . .	70
5.6.1	Reducing transaction costs . . . . .	71
<b>6</b>	<b>Proof of concept</b>	<b>73</b>

---

6.1	Platform showcase . . . . .	73
<b>7</b>	<b>Discussion</b>	<b>81</b>
7.1	Issues and possible solutions . . . . .	81
7.1.1	Privacy for the peer review process . . . . .	81
7.1.2	Privacy for the rating process . . . . .	82
7.1.3	Achieving Accountable Anonymous Reviews . . . . .	84
7.1.4	Other Problems . . . . .	87
7.2	Monetary Impact . . . . .	88
7.3	Impact in the scientific community . . . . .	88
7.4	Platform legitimization . . . . .	89
<b>8</b>	<b>Conclusions and future work</b>	<b>91</b>
8.1	Concluding remarks . . . . .	91
8.2	Future Work . . . . .	92
8.3	Conferences and Papers . . . . .	93
8.4	Acknowledgements . . . . .	94
8.5	Conclusiones finales . . . . .	95
8.6	Trabajo futuro . . . . .	96
8.7	Conferencias y artículos académicos . . . . .	97
8.8	Agradecimientos . . . . .	98
	<b>Bibliography</b>	<b>101</b>



# List of Figures

2.1	Digram of the diferent network architectures. . . . .	19
2.2	The change of state of Bitcoin accounts done by transactions. . . . .	22
2.3	Blockchain representation diagram. . . . .	23
2.4	Existing cryptocurrencies represented by purchase-sale volume.. . . .	24
3.1	Image of the value proposition canvas after the session. . . . .	34
3.2	Remix smart contract web compiler. . . . .	40
3.3	Transactions mined by testrpc in a test network. . . . .	41
4.1	Sequence diagram of platform interaction. . . . .	48
5.1	Architecture diagram of a node with IPFS and an Ethereum light client. . . . .	54
5.2	First prototype's diagram of the contracts structure. . . . .	56
5.3	Second prototype's diagram of the contracts structure. . . . .	60
5.4	Use of Ethereum's addresses in Decentralized Science. . . . .	62
5.5	Researcher's transactions list. . . . .	63
5.6	Activity diagram of sending a paper. . . . .	65
5.7	Activity diagram of assigning a reviewer. . . . .	66
5.8	Activity diagram of accepting a review. . . . .	67
5.9	Activity diagram of sending a review. . . . .	68
5.10	Activity diagram of rating a review. . . . .	69
6.1	Mock-up of the possible interactions of the platform's front-end. . . . .	74
6.2	The front page of the platform. . . . .	75
6.3	Register journal page. . . . .	75
6.4	Journal page. . . . .	76
6.5	Submission page. . . . .	77

6.6	Paper page. . . . .	78
6.7	Author's page. . . . .	79
6.8	Journal's management page. . . . .	80
6.9	Assign modal window from journal's management page. . . . .	80
7.1	Ethereum transaction fees evolution. . . . .	87



# List of Tables

5.1	Requirements of sending a paper. . . . .	65
5.2	Requirements of assigning a reviewer. . . . .	66
5.3	Requirements of accepting a review. . . . .	67
5.4	Requirements of sending a review. . . . .	68
5.5	Requirements of rating a review. . . . .	69
7.1	Different configurations to rate a review . . . . .	83



## Part I

# Introduction and previous considerations

This part aims to introduce the reader to the circumstances in which this project has arisen. It consists of an introduction and an exploration of the motivations of the project. Further on, it establishes a background of the areas to which this platform intends to affect. Finally, this part concludes describing the methodologies and technologies used for the implementation.



# Chapter 1

## Introduction, objectives and document structure

*Who Watches the Watchmen?*

Watchmen - Alan Moore

### 1.1 Introduction

Scientific research nowadays is based on publishing in journals with a high **impact factor** (*IF*)[1]. A researcher's career can be measured depending on the number of papers published in these journals. There are different impact factors that determine the quality of a journal. One of the most well-known is the Journal Citation Reports (*JCR*), an indicator that represents, for each indexed journal, the relation between the number of citable items and the number of citations they get. This *IF* is calculated every year and it is divided in four quartiles that determine the ranking of a certain journal, meaning that a journal in the first quartile (Q1) has higher impact factor than one in the second (Q2), third (Q3) or fourth (Q4) quartiles<sup>1</sup>. *JCR* was originally an evolution of the Science Citation Index, born in 1955 [2] and nowadays managed by a company called Thomson Reuters<sup>2</sup>. *Could the scientific community rely upon themselves rather than on a private company to decide the quality of an academic journal?*

One of the problems in academia is the publishing obsession. Ideally, a research must achieve publications in indexed journals. This idea causes, in some cases, to

---

<sup>1</sup><https://jcr.incites.thomsonreuters.com/>

<sup>2</sup><https://www.thomsonreuters.com>

yield to the demands of the reviewers or the editors of a journal, potentially reducing the originality or novelty of a research paper [3]. Nowadays, universities are increasingly pressuring researchers to publish papers in journals with high impact metrics, forcing them to focus their research projects on generating publishable material [4]. *Is there a better way to make a career in academia rather than pushing researchers to generate redundant or unoriginal publications?*

Science publication and peer review are build on a paper-based paradigm, with only a few changes in the last centuries [5]. The mentioned *peer review* is the process to decide if a paper is suitable to be published. A group of “experts” in a certain subject review the paper and issue this verdict. Moreover, this process has been criticized in several aspects such as: 1) The reviews are not always entirely objective, since there are cases of unfavorable reviews due to gender causes, especially in scientific fields [6]. 2) The review time of a paper is usually long, causing the process of academic research to be quite slow [7]. 3) The reviews not always ensures the quality of a paper, and cannot be used to decide if a research is good or not [8]. *Would it be possible to find better alternatives to improve this process so that it is more honest, fair or fast?*

The economical benefits of scientific distribution are centralized in a few publishers, nor the authors, the reviewers or the readers get money from it. Despite the development of the Internet enabled the proposal of alternatives for science dissemination [9] and evaluation [10], these benefits are still concentrated in the mentioned publishers. The reduction of distribution costs enabled a wider access to scientific knowledge, and questioned the role of these traditional publishers [11]. Nevertheless, universities normally have to take charge of the costs to access the papers published in these journals, paying, in some cases, very high fees [12]. On the other hand, Open Access and Open Science movements have successfully reduced the economic cost of accessing knowledge to readers [13]. However, it has not successfully challenged traditional publishers’ business models [14], who are now combining charging readers and charging authors [15]. *Could the scientific community build a system to decentralize the benefits of science publication and reward the authors and the reviewers for their work?*

Editors who assign the review of a paper to a series of reviewers have to rely on them beforehand. Thus, limiting the spectrum of fields that can be reviewed to the fields in which those reviewers are experts. In order to broaden this spectrum, the internet offers the possibility to find experts in all kind of fields all around the world. Nevertheless, when it comes to trust total stranger, it should be a system in which

anyone can rely to find trustful people. Reputation systems are the solution to these problems, since they offer a good first impression about an unknown person [16]. If an editor wants to contact new reviewers with certain fields of expertise, she could use a researcher-based reputation system to do so. Editors usually find new reviewers by reference or by the number of publications they have, but there is no easy way to predict her quality from factors like expertise and training [17]. *Could there be a system in which reviewers get rated based on their reviews and build up their reputation based on good practices and helpful reviews?*

Finally, peer review has suffered multiple criticism, and yet only marginal alternatives have gathered success [18]. The literature provides multiple proposals around open peer review [19], and proposals of reputation networks for reviewers [20]. In fact, a start-up, Publons<sup>3</sup>, provides a platform to acknowledge reviews and open them up. This work aims to provide some insights and help to alliviate many of the problems mentioned before trying to answer the questions asked previously.

## 1.2 Objectives

### Objective 1: Create a decentralized platform for science publishing

This work proposes the design and development of a platform for open science. This platform should allow its users to do the following interactions: submit papers, assign reviewers, submit reviews and rate the reviews. This will be achieved using decentralized technologies such as Ethereum (a decentralized ledger where each interaction is recorded in a public blockchain) and IPFS (a distributed file system).

The platform should be accessible through a web page in a custom server acting as a bridge between the two technologies mentioned before. Also should give its users the possibility to download the source code and running it locally, as it is a fully decentralized platform.

All the information held by the platform will be free and public, granting the possibility to each user to see all the journals, papers, reviews and ratings.

### Objective 2: Create a reputation system for reviewers

Reviewers rarely get credit for their work. Journals and conferences look for *volunteers* to review the papers submitted to those, but normally reviewers remain

---

<sup>3</sup><https://publons.com/>

anonymous. To achieve this recognition, this work proposes a reputation system for reviewers, in which every review they submit can be rated. This rating builds up a score for each one determining the *reputation* of a reviewer.

The platform should allow users to send ratings to an specific review. This rating will be saved and will be used to calculate the score of the reviewer that submitted that particular review.

With this idea, this work pretends to *foment* good and fair reviews, and avoid bad ones, trying to mitigate the possibility of unfair reviews due to gender causes, research rivalry or ignorance of a subject.

### Objective 3: Analyze the platform

After developing a functional prototype with which to perform tests, this work also aims to analyze the behavior of the platform based on the entire process of scientific publication, from the first paper submission to the final publication.

Tests will be carried out to calculate price estimates of the entire process, execution times, resistance to large amounts of information and monetary impact within the academic community.

Finally, comparisons of the results obtained will be shown based on the current publication process.

## 1.3 Document Structure

In this work there will be the following sections:

### Part 1: Introduction and previous considerations.

- **Project background:** Background about the scope of the project and what technologies are trying to change the current publication systems and how they are affecting the scientific community.
- **Methodology and Technology:** Methodology followed during the realization of this work, and the technologies used to implement the platform's architecture.



**Part 2: Decentralized Science.**

- **Platform description:** Platform general description, featuring its main strengths regarding the current platforms and explaining how it works and what is the expected behavior if it is widely used in the future.
- **Architecture:** Technical description of the platform, including the HTML and JavaScript architecture (front-end), the definition of the smart contracts' infrastructure (back-end) and the process followed to reduce the interaction costs.
- **Product:** Proof of concept of the platform, how it works and how the users can interact with the blockchain and with the p2p file network.
- **Discussion:** Results obtained after the realization of the work proposed in this project, how it will affect the scientific community and how to measure the potential impact of the platform.
- **Conclusion and future work:** Implications of this work in the scientific community and the next steps to follow to create an ecosystem of autonomous publication systems, without the need of middlemen such as journals or editors, and a proposal of a future Ph.D. about this subject.



## 1.4 Introducción

La investigación científica hoy en día se basa en publicar en revistas con alto **índice de impacto** [1], la carrera de un investigador se puede medir en función del número de artículos académicos que ha publicado en estas revistas. Hay diferentes índices de impacto que determinan la calidad de una revista. Uno de los más conocidos es el *Journal Citation Reports* (JCR), un indicador que representa, para cada revista indexada, la relación entre el número de objetos citables y el número de citas que obtienen. Este factor es calculado cada año y está dividido en cuatro cuartiles que determinan el ranking de cada revista. Esto quiere decir que una que esté en el primer cuartil (Q1) tiene mayor índice de impacto que las de los cuartiles dos (Q2), tres (Q3) o cuatro (Q4)<sup>4</sup>. *JCR* fué originalmente una evolución de el llamado *Science Citation Index*, que nació en 1955 [2] y que hoy en día es controlado por una compañía privada llamada “Thomson Reuters”<sup>5</sup>. *¿Podría la comunidad científica depender de si misma en vez de en una compañía privada para decidir la calidad de una revista académica?*

Uno de los problemas en el mundo académico es la obsesión por publicar. Idealmente, una investigación tiene que conseguir muchas publicaciones en revistas indexadas. Esta idea causa, en algunos casos, que los autores de los artículos tengan que suplir las exigencias que les imponen los revisores y los editores de estas revistas, reduciendo potencialmente la originalidad y la novedad de un artículo de investigación [3]. En la mayoría de los campos de investigación, las universidades presionan cada vez más a los investigadores para publicar en revistas de alto impacto, forzando a estos a centrar su investigación en generar material publicable [4]. *¿Podría existir una forma mejor de hacer carrera en el mundo de la academia en vez de forzar a los investigadores a generar material redundante o poco original?*

La publicación científica y el proceso de revisión por pares están contruidos sobre un paradigma basado en artículos, con pocos cambios en los últimos siglos [5]. El mencionado proceso de *revisión por pares* es el que se utiliza hoy en día para decidir si un artículo académico es apto para ser publicado o no. Un grupo de “expertos” en una materia en concreto revisan el artículo y emiten un veredicto. Pero este proceso ha sido criticado en varios aspectos tales como: 1) Las revisiones no siempre son del todo objetivas, ya que existen casos de revisiones desfavorables por causas de género, especialmente en los campos de ciencias puras [6]. 2) El tiempo de revisión de un

---

<sup>4</sup><https://jcr.incites.thomsonreuters.com/>

<sup>5</sup><https://www.thomsonreuters.com>

artículo suele ser largo, provocando que el proceso de investigación en algunos casos se ralentice [7]. 3) Las revisiones no siempre garantizan la calidad de un artículo académico, y no pueden ser utilizadas para decidir si una investigación es buena o no [8]. *¿Podrían explorarse alternativas para mejorar este proceso para que sea más honesto, justo o rápido?*

Los beneficios de la distribución científica están centralizados en unas pocas editoriales, ni los autores, los revisores o los lectores obtienen dinero de este sistema. Además, a pesar de que el desarrollo y la expansión de Internet han proporcionado nuevas formas de diseminación [9] y evaluación [10] para el proceso de publicación, los beneficios siguen concentrados en dichas editoriales. La reducción de costes de distribución ha causado un mayor acceso al conocimiento científico, y por ello se ha cuestionado el papel de las editoriales tradicionales en este sistema [11]. Sin embargo, las universidades normalmente asumen los costes de acceso a los artículos publicados en estas revistas a través de suscripciones algunas veces injustas [12]. Por otra parte, los movimientos *Open Access* y *Open Science* han reducido con éxito estos costes para los lectores [13]. No obstante, esto no ha sido suficiente para repartir y descentralizar los beneficios del modelo de negocio de las editoriales tradicionales [14], las cuales cobran a los autores en vez de a los lectores [15]. *¿Podría la comunidad científica construir un sistema para descentralizar los beneficios del proceso de publicación y recompensar tanto a los autores como a los revisores?*

Los editores de una revista que necesiten asignar revisores para los artículos que reciben, tienen que confiar en ellos de antemano. Esto puede limitar el espectro de campos para los que un editor pueda encontrar revisores. Pese a ello, internet ofrece la posibilidad de encontrar a gente en todo el mundo, pudiendo encontrar revisores expertos en todo tipo de materias. Sin embargo, cuando se trata de confiar en una persona desconocida, debería existir un sistema en el que cualquiera pueda confiar para encontrar revisores de calidad. Los sistemas de reputación son la solución a este problema, ya que ofrecen una primera impresión de una persona basada en las opiniones de otras que hayan interactuado con ella [16]. Si un editor de una revista quiere ampliar su plantilla de revisores necesitaría contactar con nuevas personas, las cuales pueden ser desconocidas. El problema es que no es fácil determinar la calidad de un revisor basándose en su experiencia [17] por lo que un sistema de reputación podría ser útil para encontrar buenos revisores sin tener que conocerlos de antemano. *¿Se podría construir un sistema en el que los revisores son puntuados para conseguir reputación basada en lo buenas o malas que sean sus revisiones?*

## 1.5 Objetivos

### Objetivo 1: Crear una plataforma descentralizada para la publicación científica

Este trabajo propone el diseño y el desarrollo de una plataforma para *Open Science*. Esta plataforma debería permitir a los usuarios: enviar artículos, asignar revisores, enviar revisiones y puntuar estas revisiones. Para ello se utilizarán tecnologías descentralizadas como Ethereum (una plataforma distribuida en la que cada interacción es grabada en una base de datos pública) e IPFS (un sistema de archivos distribuido).

La plataforma será accesible a través de una web en un servidor local que actuará como interfaz ocultando los detalles de las dos tecnologías mencionadas. Además, al ser distribuido, se ofrecerá a los usuarios el código fuente para poder ejecutar el sistema en un nodo local.

Toda la información de la plataforma será pública y gratuita, dando la posibilidad de consultar las revistas, los artículos académicos, las revisiones y la reputación de todos los revisores.

### Objetivo 2: Crear un sistema de reputación para revisores

Los revisores raras veces obtienen reconocimiento por su trabajo. Las revistas científicas y las conferencias normalmente buscan *voluntarios* para llevar a cabo el proceso de revisión, pero en la mayoría de los casos, los revisores permanecen anónimos. Para conseguir este reconocimiento, este trabajo propone el desarrollo de un sistema de reputación de revisores, en el que cada revisión puede ser puntuada. Esta puntuación es asociada a cada revisor y determina la *reputación* que tiene.

La plataforma debe permitir a los usuarios calificar cada una de las revisiones que obtienen los artículos académicos. Esta calificación servirá para calcular la puntuación de cada revisor dentro de la plataforma, haciendo que los revisores que realicen buenas revisiones tenga buena reputación y los que no, mala.

Con esta idea, este trabajo pretende *fomentar* revisiones buenas y entregadas a tiempo y disuadir a aquellos revisores que no estén dispuestos a esto, mitigando en la medida de lo posible revisiones injustas debido a causas de género, rivalidad de investigación o desconocimiento de una materia.

### Objetivo 3: Analizar la plataforma

Después de desarrollar un prototipo funcional para realizar pruebas, este trabajo propone analizar el comportamiento de la plataforma teniendo en cuenta en el proceso completo, desde el envío del primer artículo hasta su publicación final.

Se realizarán varios test para calcular estimaciones del coste de todo el proceso, tiempos de ejecución, resistencia a grandes cantidades de información e impacto monetario en la comunidad científica.

Finalmente se ofrecerán conclusiones de los resultados obtenidos para analizar la viabilidad de la implantación de esta plataforma frente a los sistemas actuales.

## 1.6 Estructura del documento

Este trabajo dispone de los siguientes capítulos:

### Part 1: Introduction and previous considerations.

- **Project background:** trasfondo en el cual incide el proyecto y diferentes tecnologías actuales para cambiar el proceso actual de publicación científica.
- **Methodology and Technology:** explicación de las metodologías y tecnologías utilizadas durante el desarrollo de todo el proyecto.

### Part 2: Decentralized Science.

- **Platform description:** descripción general de la plataforma, presentando el funcionamiento esperado y las principales ventajas del diseño.
- **Architecture:** descripción técnica de la plataforma, incluyendo la arquitectura del *frontend*, la definición del funcionamiento interno y el proceso seguido para reducir los costes de interacción.
- **Product:** prueba de concepto de la plataforma, mostrando un prototipo funcional que interactúa con las tecnologías distribuidas mencionadas previamente.
- **Discussion:** resultados obtenidos después del desarrollo de la plataforma y cómo estos afectarían a la comunidad científica.
- **Conclusion and future work:** implicaciones de este trabajo, observaciones finales, y propuestas para un futuro proyecto de doctorado.

## Chapter 2

# Project background

*The future is already here – it's just not evenly distributed.*

William Gibson

### 2.1 Socio-cultural background

#### 2.1.1 Publication systems

The methodology of scientific publications creation was established in 1620 [21], when Francis Bacon established certain steps to elaborate what we know today as scientific papers. But it was 45 years later, in 1665, when appeared what we consider the first scientific journal: *Philosophical Transactions of the Royal Society* [22]. At that time, editors were the ones who had to carry out the review of the papers that would be published in these journals.

It was around 100 years later when an alternative system was adopted, instead of editors doing the work of reviewing all the papers, this would be done by a group of experts in a certain field, deciding if each paper reviewed was good enough to be published or not. This is the beginning of the process known today as “peer review” [5].

But scientific publication as we know it today was settled down in the 19th century, with the actual peer review process [23], establishing the guidelines of the paper-based paradigm that we have in science nowadays.

Determining the quality of a scientific paper is difficult, but today we have different ways to do so. To be able to estimate this quality, usually there are two

approaches: before and after publishing.

In the **peer review process**, researchers in a certain field evaluate a paper's quality, commonly implying its eligibility to be published. These reviewers read the paper and submit a review and an "acceptance score" representing if they think the paper should be accepted. Normally these researchers are unknown to the authors and the reviews are made anonymously (*blind review*), sometimes reviewers also do not know who the authors are (*double blind review*), and in rare cases, both the authors and the reviewers are public (*open review*). Therefore it could be considered that this process is an indicator of the quality of a paper before being published [24].

Another way to determine a papers quality is after it is published, obtaining the number of citations it gets over time [25]. Normally one with a higher number of citations is considered to have better quality than one with lower number of citations. This way of analyzing quality gives rise to the generation of indexes that determine how good a researcher is. For example, the H index that assigns a numerical value to each researcher based on the articles she has written and the citations he has received [26]. Regarding these metrics, several alternatives have been explored, since determining whether a researcher is good or not through a number is not always correct [27]. However, in most cases, it is possible to predict through this index if the quality of future research papers will have good quality or not [28].

The H index could be one of the most used factors to measure the impact that has a researcher. There are a series of similar indexes not to measure the impact of a researcher, but the impact of a scientific journal. Also resorting to the citations rate, journals size their quality with these indexes. JCR (journal citations rate) [1] is one of the most widely used and recognized in the scientific community (see chapter 1). Researchers who seek to get their work recognized should publish papers in journals with high JCR.

Today's scientific research system is based on publishing papers in journals and conferences, as the papers of an academic research that are published in any other medium are part of the called "grey literature", that usually are not taken into account in the research process [29].

People looking to make a career in academia should bear in mind that one of the priorities is to publish the maximum number of articles in high-impact journals, as discussed above. In Spain for example, entities such as the ANECA <sup>1</sup>, determine if a person can teach at the university based on a number of factors. One of the most important is the number of papers published in journals with high JRC that person

---

<sup>1</sup>[www.aneca.es/](http://www.aneca.es/)



has. In this way a need is generated, which certain actors take advantage of within this system.

Many of the journals with high impact factors belong to **publishers**. In the past, these publishers were in charge of producing, printing and distributing the editions of the journals when they began to emerge. Many major publishers such as Willey-Blackwell<sup>2</sup>, Elsevier<sup>3</sup> or Springer<sup>4</sup> have been around since the beginning of the 19th century. Nevertheless, the digital age has meant that copying a document, which used to be an expensive process, now has a very low cost. Even so, publishers continue to profit from this system, acting as intermediaries between the people who create science and those who consume it [14]. In the era in which the replication of information is not a cost, the scientific publication process could be migrated to fairer and more honest systems.

### 2.1.2 Alternative publication systems

Publication systems, as seen on the previous section, form an oligopoly of a few, concentrating the benefits of the industry. However, there are some attempts to change this paradigm on behalf of science dissemination.

**Open Access** is a concept referring to all the research material that is free of cost for the readers. This concept also involves various conditions for something to be considered “open access” [30]:

- **Freely accessible online:** Any user can get a copy of the paper online without any cost, including reviewed papers and unreviewed papers (preprints).
- **Lax copyrights:** Any user can download, print, search, copy and link to the full text or any part of these papers.
- **Public support:** Everyone should be able to access to all the material related to the paper, and should be uploaded to a public repository supported by an academic institution or other well-established organization that seeks to enable open access.

But as mentioned in the chapter 1, in most *open-access* journals, the costs lie with the authors rather than the readers [14; 15], making unaffordable for some researchers or universities the possibility to publish in these journals.

---

<sup>2</sup><https://www.wiley.com>

<sup>3</sup><https://www.elsevier.com/>

<sup>4</sup><http://www.springer.com/>

**Mega-journals** (or Multi-journals) [31; 32] combine multiple journals into a single journal, allowing the publication of open-access papers, which have gone through a peer review process. The first journal to adopt this idea was the *PLOS ONE* Journal<sup>5</sup> of the project *Public Library of Science*. This project aims to create a library of scientific journals under the values of open access and creative commons licenses. As a result of the success of the *PLOS ONE* journal, other publishers have started their own mega-journals. Featuring alternative impact metrics, reusability of figures and data, post-publication discussions and portable reviews from other journals [33].

The **continuous publication** model is based on publishing individual papers once they are reviewed and accepted, migrating from the previous issue-based model [34]. This method is seen as an alternative for open-access journals as it speeds up the publication process [35]. *Decentralized Science* adopts this model by design (explained in section 5.1) as the platform automatically publishes papers that meet certain preconditions, such as getting at least two out of three positive reviews.

**Preprints** are scientific papers that have not yet gone through the peer review process [36]. Formerly, the preprints that were sent to the journals were private, and only accessible by the editors and assigned reviewers. Nevertheless, nowadays it is common to publish a preprint before sending it to a journal, uploading it to specialized platforms like arXiv<sup>6</sup> or Preprints<sup>7</sup> [37]. Moreover, there is a correlation between the upload of a preprint and early citations after the publication of the paper [38]. This approach is a possible solution to the cold-start problem that papers of new researchers who enter the academic career [39].

Social networks have also made a dent in the academic world, creating platforms to contact other researchers and encouraging them to share their papers. These platforms allow users to have a public profile, in which they can add relevant information about their career: research fields, interests, papers published, etc. The users also have the possibility to connect with other users and add them as contacts or fellow researchers. In addition, these platforms allow their users to upload scientific papers (reviewed or unreviewed), granting the possibility to share them through the platform's community. Some of the well-known are Research Gate<sup>8</sup>, Mendeley<sup>9</sup> or Academia<sup>10</sup>. But despite the good intentions of these platforms, many of the jour-

---

<sup>5</sup><http://journals.plos.org/plosone/>

<sup>6</sup><https://arxiv.org/>

<sup>7</sup><https://www.preprints.org/>

<sup>8</sup><https://www.researchgate.net>

<sup>9</sup><https://www.mendeley.com>

<sup>10</sup><http://academia.edu>

nals demand the copyright of the papers they publish, preventing the authors from sharing them through these services.

In the current scientific publication process, an important part of the work is done by reviewers, who remain anonymous for both the authors and the rest of the scientific community. Decentralized Science proposes, among other things, to make a reputation system of reviewers so that they can obtain the credit and recognition of carrying out good reviews, and be penalized otherwise.

### 2.1.3 Reputation systems

**Trust** in an individual is hard to acquire. Normally trust can not be bought, sold or exchanged. When choosing whether to trust someone or not, having the opinion of other people about that individual can be helpful. If these opinions about each individual are stored in a system which can be consulted by anyone, it is called a “reputation system”.

A reputation system is a technology that allows the users to trust third parties inside the same system, without prior knowledge of each other. The reputation system collects, adds and distributes the comments received from the behavior of the participants, based on interactions with each other. This idea was born in the early 2000s in which the use of the internet spread throughout the world, and people needed to interact and trust strangers [16].

The “reputation” is normally a value that indicates how much confidence the community has in a user. This reputation is gained through interactions with the rest of the people who use the service or platform that implements the reputation system. Sometimes these platforms offer *privileges* to people who have a certain level of reputation, unlocking certain actions that can only be performed at a certain threshold.

The basic idea of reputation systems is to give users the possibility to rate the interactions that occur between them. One of the pioneering platforms to implement this system was Ebay <sup>11</sup>, a second-hand buying and selling website. In it, each vendor had a reputation based on whether previous transactions had been honest or not. Each user who purchased products from that seller had the opportunity to rate the purchase. In this way, all those vendors who tried to rip off users immediately received a bad reputation score.

---

<sup>11</sup><https://ebay.com>

Many of the large internet communities such as Stackexchange<sup>12</sup> or reddit<sup>13</sup> have their own reputation system. Reputation systems behavior may vary depending on the platform [40], but the most usual is the one where users get a score based on certain interaction with the community.

Reputation systems also have a wide niche in e-commerce webs such as Ebay, as mentioned before, or Amazon<sup>14</sup>, in which people pay for a product sold by an unknown vendor. There must be a previous trust in the vendor before buying any product, so a reputation system offers a score given by other users that encourages you to trust or not that certain seller [41].

These systems vary widely in scope, such as one for peer-to-peer computing [42], vehicle ad-hoc [43], web services [44] and even Wikipedia [45]. All of them are based on an exchange of trust between users of these services.

This same concept was intended to be implemented using a token as a **trust unit**, which users exchanged as a sign of trust deposits among them [46], but this project is still in its design phase and a functional platform has not yet been developed.

Nevertheless, reputation systems also have problems when it comes to defend the users from attacks to individuals [47] and unfair ratings [48]. The fact that a platform is implemented in the blockchain implies that all interactions are public and auditable. Everyone can see all votes and ratings, so it is a system in which there is no anonymity. This could allow dissuading the mentioned problems since the users who carry out unjust directing or rating attacks are exposed publicly in the network. Nevertheless, this also raises concerns about privacy and anonymity.

Nowadays the most widely used reputation systems are based on a five-star rating [49]. This reputation system is present in many online platforms in some of the mentioned before and others like Tripadvisor<sup>15</sup>, AliExpress<sup>16</sup>, Google Play<sup>17</sup> and other large platforms in which products and services are voted by users.

---

<sup>12</sup><https://stackexchange.com/>

<sup>13</sup><https://www.reddit.com/>

<sup>14</sup><https://amazon.com>

<sup>15</sup><https://www.tripadvisor.com/>

<sup>16</sup><https://aliexpress.com/>

<sup>17</sup><https://play.google.com/>

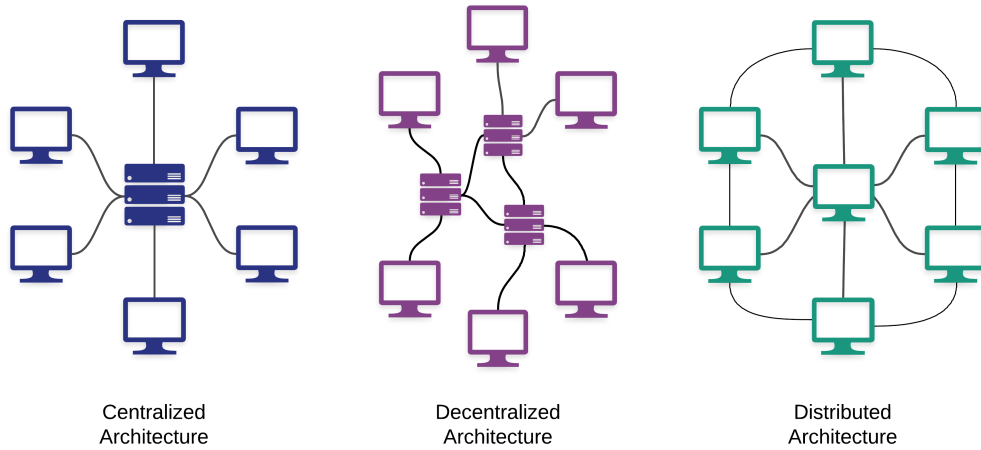


Figure 2.1: Diagram of the different network architectures.

## 2.2 Technical background

### 2.2.1 Network Architectures

Most of the services that we use everyday are on the internet. All the users or these platforms need to be connected to the network in order to use them. These services are usually offered by centralized entities to which users have to connect in order to access them. On the other hand, there are also other platforms that offer their services without centralizing all the traffic in the same site. For each one of these services, there are different architectures, which can be divided into three different groups (see figure 2.1 [50]):

- **Centralized Architecture:** All the structure is managed and controlled by a single node. This structure is vulnerable, meaning that attacks to the central node can compromise the entire network [50]. This type of architecture is used nowadays in web services in which all of its users have to connect to a centralized server. The majority of web pages and services on the internet use this architecture.
- **Decentralized Architecture:** This type of structure shows a hierarchical structure to a set of *stars* connected in the form of a larger star with an additional link forming a loop. Each *star* is managed by its community. Attacks to a single *star* can suppose a loss of communication between nodes but does

not compromise the entire system [50]. Besides, all its users can access all the data stored in all the nodes. This infrastructure is used in platforms like GNU Social<sup>18</sup>, Buddycloud<sup>19</sup> and Diaspora<sup>20</sup> among others.

- **Distributed Architecture:** This is a fully decentralized architecture and its usually called “Peer to peer” or “P2P”. All the nodes (or *peers*) share part of its own resources, and these shared resources are necessary to provide the service and content offered by the network. They are accessible by other peers directly, without intermediary entities [51]. All the data is spread among the nodes which have the same privileges in the structure. Users can control its contribution to the network and everyone can join or leave at any time without affecting the network architecture. Today this structure is used in platforms like BitTorrent<sup>21</sup>, Bitcoin<sup>22</sup> and Ethereum<sup>23</sup>

Despite the disadvantages, most sites choose a centralized structure since in many cases they are controlled by private companies. However, distributed architectures are increasingly being used by new technologies since, as explained above, they offer more advantages than centralized ones. One of the clear examples of the use of this architecture is that of cryptocurrencies.

### 2.2.2 Digital Currencies and Cryptocurrencies

In the 80s and 90s, with the expansion of the World Wide Web, several services began to be online (stores, newspapers, etc). Some of these services required monetary transactions through the internet to be used so, at this time, the concepts of electronic money or digital currencies began to emerge. One of the first forms of digital payment was Ecash [52] in 1995, an idea of a digital currency that offered a high level of privacy. Unfortunately at that time the use of the internet was not so widespread and the idea did not become popular.

Three years later, in 1998, Wei Dai published his proposal for electronic money called B-money [53] an "anonymous, distributed electronic cash system", a revolutionary idea for the time, since it added the concept of anonymity to monetary

---

<sup>18</sup><https://www.gnu.org/software/social/>

<sup>19</sup><https://buddycloud.com/>

<sup>20</sup><https://diasporafoundation.org/>

<sup>21</sup>[www.bittorrent.com/](http://www.bittorrent.com/)

<sup>22</sup><https://bitcoin.org/>

<sup>23</sup><https://www.ethereum.org/>

transactions. B-money added a layer of cryptography to secure these transactions that is why these ideas began to be called “cryptocurrencies”.

Hashcash [54] was another digital currency idea that was born in 2002 and introduced the concept of *proof of work*. A *proof of work* is a piece of data which is difficult to generate, because it is time consuming or have high cost, but easy for others to verify. Hashcash used *Hash Functions* with certain requirements to encrypt transactions of money between users.

Despite the fact that these projects were never carried out, they inspired others that would come later such as Nick Szabo’s Bit gold [55] a similar approach as B-money but using P2P protocols and PRPOW [56] an invention by Hal Finney intended as a prototype for a digital cash.

None of these currencies was implemented successfully enough until 4 years later with the arrival of the most widely used cryptocurrency today: **Bitcoin**.

### 2.2.3 Bitcoin

In 2009, the idea of a **fully decentralized currency** first emerged, when Satoshi Nakamoto, a fictional identity, published the first version of Bitcoin [57]. Decentralizing a currency implied that there was no entity that controlled it, not even banks, governments or companies. Bitcoin is controlled by all users who use it, since it uses a P2P network in which everyone can see and verify all money transactions. This technology used concepts and ideas of previous digital currencies such as B-money, Hashcash and Bit gold (see section 2.2.2).

In order to understand this technology, two important concepts must be defined: addresses and transactions.

Each user who wants to use this cryptocurrency, must first have an account, a randomly generated text string called **address**. Each address can be generated with no cost and users can have multiple addresses. These addresses act as an identifier inside the Bitcoin’s database and are used to transfer money.

Money transfers between two *addresses* are called **transactions**. Each transaction is a small data fragment containing the source address, the destination address and the amount of money transferred (see figure 2.2).

Bitcoin’s users can observe all *transactions* between all *addresses*. In order to do so, this technology stores a certain number of transactions into a data fragment, called block. Next, this block is linked cryptographically to the previous one containing all the foregoing transactions, forming a chain of interlinked data called

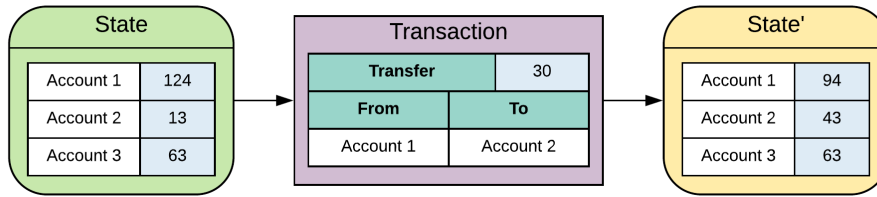


Figure 2.2: The change of state of Bitcoin accounts done by transactions.

*blockchain*. To sum up, Bitcoin’s blockchain is a large chain of data blocks containing all the transactions between all accounts since the beginning of this technology.

Bitcoin’s transactions must be verified by all peers, meaning that there has to be a consensus among all the nodes for a transaction to be completed. This solved the double-spending problem by which a dishonest actor may try to spend twice the same coin in decentralized currency systems [58].

Bitcoin also introduces incentives to maintain the security of this ledger, both rewarding nodes that contribute computational power for the security of the network, and requiring at least half of the computing power of the network to alter the state of the blockchain (thus, the blockchain is secure if at least half of the computing power is provided by honest peers).

To deepen more into Bitcoin’s inner functioning, this technology injects money into the network using the idea of *proof of work* or POW.

POW consists in all users competing with each other to encrypt a data block with certain restrictions. The winner notifies its neighbors with the data block successfully encrypted and gets its reward in money [59].

Specifically, to encrypt a block, peers need to gather a number of pending transactions into a data structure. Next, they add a random number (called “nonce”) to that structure so that when performing a hash function of the whole block has a certain number of leading zeros. Once a block is created, it is used as reference to the next block of data, adding the hash of the previous block to the next one [60] (see figure 2.3).

This system makes it practically impossible to falsify a transaction in the blockchain, since all nodes need to reach consensus. However, this system is still vulnerable to attacks aimed at specific users, such as man in the middle attacks [61].

This technology enabled a new wave of decentralization of applications such as



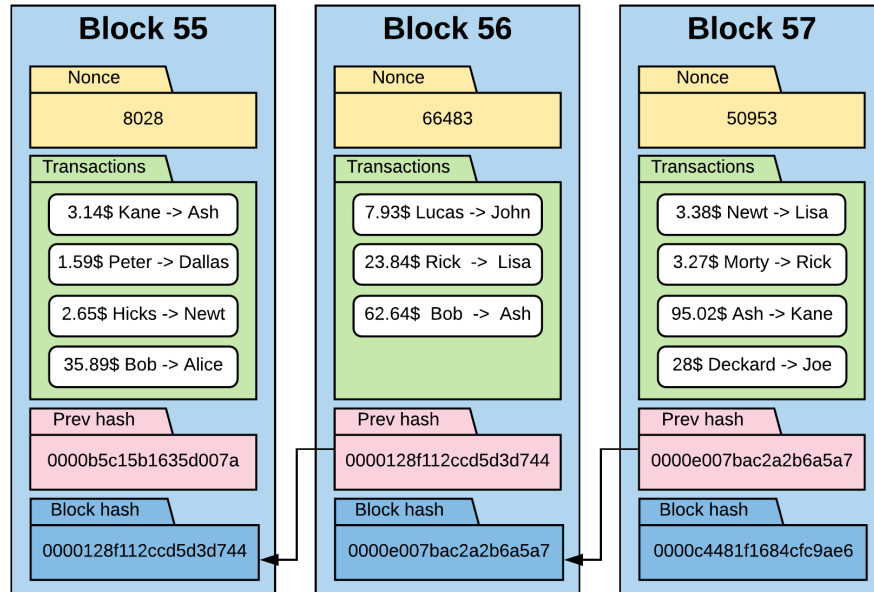


Figure 2.3: Blockchain representation diagram.

domain name registries [62] or microblogging platforms [63]. A second wave of blockchain based decentralization was started by Ethereum [64], as described below.

#### 2.2.4 Ethereum

Ethereum [64] is a very novel technology that allows the creation of distributed applications that run in an arbitrary large and trust-less network of nodes. Ethereum is based in the Bitcoin's blockchain technology, a public database where everyone can watch all transactions.

Using this idea, Ethereum uses its own blockchain (see section 2.2.3) to deploy and execute fragments of code in a distributed network. These fragments of code are called “smart contracts” and they are uploaded to the blockchain in order to be executed.

Working very similar to Bitcoin's blockchain, peers have to reach consensus in each smart contract execution, making a smart contract source code almost impossible to hack.

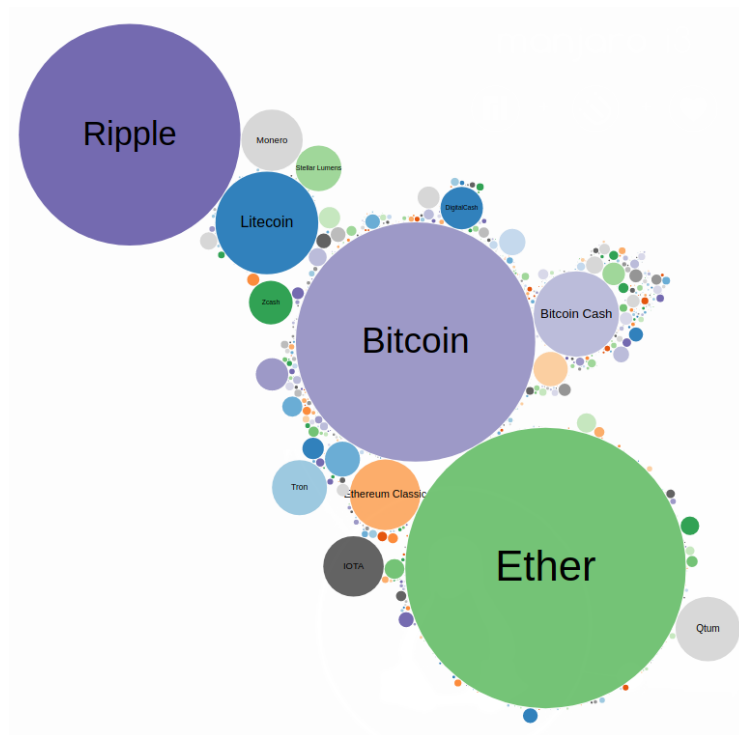


Figure 2.4: Existing cryptocurrencies represented by purchase-sale volume..

Apart from this, Ethereum uses its own cryptocurrency called “Ether”. This currency not only works like Bitcoin, to exchange money between users, but to *fuel* the smart contracts’ code execution, running its inner functions for a small amount of *Ether*.

### Smart Contracts

Smart contracts are written in a programming language called Solidity<sup>24</sup> provided by the Ethereum’s developers. This language is called contract-oriented and it was influenced by C++, Python and JavaScript. Solidity offers the possibility to create a wide range of decentralized applications in the blockchain in which users do not have to trust a centralized organization.

These contracts also have the capacity to store and transfer money, making them the perfect tool to implement a wide range of decentralized applications like: gambling games [65], voting systems [66], crowdfunding [67], prediction markets [68],

<sup>24</sup><http://solidity.readthedocs.io/en/develop/>

transparency systems [69] and so on. Today it is the most exchanged currency within cryptocurrencies <sup>25</sup> (see figure 2.4).

Smart contracts offer us a framework to design distributed platforms like the one proposed in this work, in which all transactions that interact with the scientific publication process can be cryptographically verified (see section 5.2).

### Addresses and Transactions

As in Bitcoin's network, Ethereum's user accounts are called **addresses** and the transfers between them are called **transactions**. The main difference between Ethereum and its predecessor is that an address can also be assigned to a smart contract. Thus, there are two types of addresses within the Ethereum network:

- **Personal addresses:** These are the addresses of the users who want to interact with the Ethereum network. Each one has its address and a balance.
- **Contract addresses:** Once a smart contract is deployed in the Ethereum blockchain, the network generates an address and assigns it to the deployed contract. It contains a pointer to the source code, a balance with the available money, and its own internal memory where it saves all the contract's information.

Each of the addresses within the Ethereum network is unique, so they have several useful applications when it comes to develop an smart contract. Addresses can be used to identify who executes a contract as well as to restrict the use of certain contracts to a list of allowed addresses. Listing 2.1 shows an example code of a very simple smart contract. This example contains comments to explain the address restrictions mentioned before.

It should be noted that these address restrictions can also be applied to contract addresses, and may limit the execution of a certain fragment to an address associated with another contract.

Smart contracts behavior is transaction-based [70]. Once a contract is deployed in the blockchain, users (or other contracts) may send transactions to run specific functions in its code.

---

<sup>25</sup><https://cryptocoincharts.info/coins/graphicalComparison>

```
contract Example{

    // The owner of the contract
    address owner;

    constructor() public{
        // The contract assigns the owner to
        // the account that creates the contract
        owner = msg.sender;
    }

    function doSomethingPublic() public{
        // This function can be called
        // once the contract is deployed
    }

    function doSomethingRestricted() {
        if ( msg.sender != owner ){
            // If the address that calls this function is not
            // the owner the transaction is cancelled
            revert();
        }
        else{
            // This function can be called only by the owner
        }
    }
}
```

Listing 2.1: Example of an smart contract

Each transaction has a payload, containing the data required to execute the desired function. This execution has a fee called **gas** that users have to pay to the network based on how complex is the code they want to run. The **gas** amount required to execute a function inside a smart contract directly determines the code quality, as expensive transactions to smart contracts with high **gas** costs are undesirable.

Ethereum's transaction-based smart contracts have changed the paradigm of modern software development, since the priority when developing a smart contract

is to reduce these transaction costs for each interaction [71]. For this reason, the development of this project is based on the programming incremental prototypes to be able to perform tests in a fast way (see section 3.1.4).

## 2.3 Peer review and publishing software

Nowadays, journals and conferences manage the submission process digitally. While conferences commonly use a generic software (usually web-based open source systems), journals normally have its own management system. These platforms allow editors to manage submissions and peer-review processes, as well as review assignment and notification sending.

**Event Management Systems** (EMS) or *Conference Management Systems* are wide range of software products that are used in the management academic conferences [72].

**EasyChair** [73] is perhaps one of the best known, a web-based EMS widely used by the community with more than 65202 conferences served <sup>26</sup>. This system provides the following tools: 1) paper submission; 2) review assignment; 3) email notifications to authors, reviewers and conference chairs and 4) preparation of the conference proceedings.

**OpenConf** [74] is another EMS that offers the same tools as EasyChair. It lacks of project management features and it is only suitable for conference, workshop or seminary events. OpenConf has a different licenses for its users<sup>27</sup>: a free but limited one for communities or a “Professional Edition” with extra features like web and mobile integration.

**AEMS** [75] is an advanced event management system that features useful tools like project management integration. This platform main objective is to combine the functionality of previous EMS and improve them, offering its customers tools to manage activities, resources and costs associated with an event.

In the other hand, journals normally do not have to take charge of organizing a conference or a workshop, so they do not have to worry about things like schedules or conference chairs. Authors submit their papers, papers are assigned to reviewers and papers are accepted for publication or rejected. To carry out the monitoring of this process there are several platforms called **Editorial systems** (ES) [76].

**Evisé** is a web-based ES used by the publisher “Elsevier” [77] to manage the edi-

<sup>26</sup>According to <https://easychair.org/>

<sup>27</sup><https://www.openconf.com/editions/>

torial process. This platform allows users to have a profile inside Elsevier’s database used to sing into this publisher’s journals. It also has editor oriented tools like one to search for, invite, and manage reviewers from a single screen or one to create and manage personal customized decision letters among many others.

**MySpringer** [78] by “Springer Nature” is another ES used to access the contents inside this publisher’s system. It is suitable both for authors, in which they can view its publications and manage submissions made, and for editors, who can access to a variety of tools similar to Evise’s platform. Furthermore, users can link an account in Overleaf<sup>28</sup>, a collaborative online L<sup>A</sup>T<sub>E</sub>X editor, to submit papers to the any Springer’s journal.

**Open journal systems** [79] is an open software designed to facilitate the publishing process. This project was created by the Public Knowledge Project<sup>29</sup> and it targets open-access online journals that want to speed up the publication processes. The system provides tools to control the whole publishing process from paper submission, through peer reviewing to the final publication issue.

All these platforms, despite being used today, are still chained to the archaic publication methodology that we currently have (see section 2.1.1). For example, the reviewers remain anonymous even after the publication of the paper, usually causing them not to receive any recognition for their work.

Nonetheless, there are initiatives to change this, such as the one proposed by **Publons** [80], a platform in which users make public all the reviews they make to the peer review process. Publons tries to eliminate the anonymity of this process, urging the reviewers to obtain recognition for publishing such revisions. However, it is not always possible to make reviews public, given that sometimes journals or conferences do not allow this type of data to be published.

Reputation systems at the moment have not had much impact on the academic world, since only a few initiatives have emerged [81], but none of them has been more popular than the social networks mentioned in the previous section.

Alternatives to these systems using decentralized technologies, in spite of their promises [82], are still in their infancy. A few proposals, none of them functional to date, have appeared recently. One of the most promising ones is **Aletheia**, a peer review proposal that tries to solve some of the peer review socio-technical problems using cryptocurrencies [83]. However, it needs a critical threshold of research community engagement, changing the actual processes and platforms, to start being

---

<sup>28</sup><https://www.overleaf.com/>

<sup>29</sup><https://pkp.sfu.ca/about/>

implemented.

Blockchain-enabled apps have also been proposed, with voting and storage of publications. This is the case of Aletheia [84], a software for getting open access papers published. This platform idea aims to use blockchain as a decentralized and distributed database as a publishing platform.

Peer review quality control through blockchain-based cohort trainings [85] have been also proposed, with the promise of transparency and decentralization using a distributed ledger. Research labs can use this training network to test their technology and reduce the risk for private investment opportunities.

Finally, some of the off-chain journals are adapting to the demands of the current scientific community like Ledger<sup>30</sup>, a cryptocurrencies and blockchain-based journal that records the publication timestamps in the Bitcoin blockchain.

Decentralized Science aims to challenge the technical infrastructure that supports the middlemen role of traditional publishers. Due to the successes of the Open Access movement, some of the scientific knowledge is today freely provided by the publishers. However, the content is still mostly served from their infrastructure (i.e. servers, web platforms). This ownership of the infrastructure gives them a position of power over the scientific community which produces the contents [86]. Such central and oligopolistic position in science dissemination allows them to impose policies (e.g. copyright ownership, Open Access prices) and concentrate profits.

This work proposes the development of a decentralized publication system for open science, improving the mentioned systems. The platform is based in the idea of *open access*, as the information held by this system is public and free. It also embraces ideas like *preprints*, as paper submissions are accessible before the peer review process, and *continuous publication*, as every time a paper is reviewed it is published automatically. The reputation system could allow journal editors to find suitable reviewers for each paper submission. Reviewers could finally get their work recognized, offering alternatives like paid reviews. This reputation system also could be a solution for some of the problems associated with the peer review process, penalizing slow, unfair or biased reviews and rewarding good and fair ones. Finally, the economic benefits of the publication process could be distributed in the scientific community, opening new forms of projects funding.

---

<sup>30</sup><https://ledgerjournal.org>





## Chapter 3

# Methodology & Technology

*The needs of the many outweigh the needs of the few*

Spock - The Wrath of Khan

### 3.1 Methodology

Decentralized Science is based on a project idea outlined in a blockchain event in September 2017. In this event we were a group of 4 developers with one month to design an idea of a platform or service using blockchain technologies to solve a particular problem. This Master's Thesis is a first approach to the software design and implementation of that project sketch.

#### 3.1.1 Project Timeline

Decentralized Science's project timeline has three different parts:

- **Pre-project:** *Blockchain for Social Impact* was an online event that took place in September 2017. The main goal of that event was to design a blockchain-powered platform to solve any social related problem, and the best one would receive funding. Me and other partners formed a team to develop the first Decentralized Science's idea within the month that the event lasted. In this period we had to make a series of deliverables every week to show that we were working on one project. In order to make this, the team met 3 times a week to design the platform using methodologies such as Brainstorming and Value Proposition Canvas. The main idea of the platform and the problems it had

to solve were identified in this period. Unfortunately the project was not one of the winners of the event and the team decided to pause the project.

- **Project implementation:** Since I was the only one with programming knowledge in blockchain, I decided to continue with the project on my own. Consequently, and as my Master's Thesis, I started the project's implementation and software design from October 2017 until February 2018. To achieve this, I used a development system based on agile methodologies such as SCRUM and Extreme Programming to be able to develop incremental prototypes as a first approach to the platform. During this period I have developed two functional prototypes using different blockchain architectures. Afterwards, I have tested them to compare the two implementations in terms of performance, execution cost and compatibility with other possible platforms. These prototypes are developed under an **open source license** and the last version is available in Github<sup>1</sup>. During this period I had reunions twice a month with my directors to update the status of the project. Finally, since January of 2018 I have started this manuscript as the master's thesis memory.
- **Post-project:** From March 2018 the team decided to continue working in the project's idea, trying to find supporters and funding to develop a final platform. Decentralized Science has been presented in three European conferences on blockchain in which several projects were discovered in the same field: ETHCC2018 (Paris), SPOBC2018 (Vienna) and PEERE2018 (Rome). In addition, Decentralized Science has formed a group with other similar projects called Open Science Ecosystem with more than 30 projects regarding open science around the world. Currently the team are sending scientific papers to different conferences to validate the ideas of this platform with the scientific community. One paper has been published in the PEERE2018 conference and another has been accepted in a CORE-A conference so far (see section 8.7). This papers and the feedback received by the community allowed me to improve the platform source code and complete this manuscript.

### 3.1.2 Brainstorming

Brainstorming was born as a method to increase creativity in groups and organizations. There are only few rules on this method: do not criticize any of the given

---

<sup>1</sup>see <https://decentralized.science>

ideas, quantity is desired over quality, try to combine suggested ideas and give all the ideas that come to mind, no matter if they are possible or not [87].

This method is used nowadays in companies and work groups as part of the process of the creative process of a product, although there are some critics about brainstorming and in some cases instead of encouraging creativity, inhibits it [88; 89]

As part of the **pre-project**(see section 3.1.1) the team decided to use this methodology and do a brainstorming session to define what kind of problem should be solved with blockchain. Many ideas emerged and were captured into a white board without discrimination, no matter how hard or easy to implement they were.

After saying enough ideas to fill the board we filtered the ones that were impossible to achieve. Then, each one voted the best three, making a ranking of the 3-4 best projects to start working on. Some of the ideas we came up to were creating a distributed wikipedia with governance models, an application to contact people from minority groups in countries where they are persecuted collectives, a distributed and community driven NGO<sup>2</sup>, and a crowdfunding platform for *whistleblowers*<sup>3</sup>.

Finally we decided to create an approach to a distributed platform for open science which we initially called Alexandria, in honor of Alexandra Elbakyan, the creator of Sci-hub. Nevertheless we changed the name of the project to Decentralized Science because there were a similar project about distributed system for media files using blockchain with the same name<sup>4</sup>.

### 3.1.3 Value proposition canvas

A value proposition canvas is a tool to create, design and implement a product idea. This method is commonly used by businesses and entrepreneurs to find the balance between customer profile and product design, but there are other cases of use for this tool outside business scope [90; 91].

The process is divided in two parts, customer profile and value map, each of these divided in other three parts: [92]:

- **Customer profile:** The goal of this step is to identify the profile of the final user of the platform. This section is divided in three parts: 1) *Customer jobs*: things the customer are trying to get done, 2) *Customer pains*: undesired costs and situations, 3) *Customer gains*: benefits, social gains and cost savings

---

<sup>2</sup>Non-governmental organization

<sup>3</sup>A person who exposes any kind of information or activity that is deemed illegal

<sup>4</sup><https://www.alexandria.io/>



- Value Map:** This section is about what the final product has to have and what does not, and its also divided in: 1) *Product and services*: which products and services are offered that help the customer to get a job done, 2) *Pain relievers*: how the customer pains are going to be alleviated, 3) *Gain creators*: how the products and services create customer gains.

- **Customer Jobs:** Get recognition as researcher, share a research work, collaborate with peers, build a professional network, publish papers, be able to finish a researcher's PhD.
- **Customer Pains:** Slow reviews, publish in indexed journals, non-transparent

process, takes more time than other jobs, low salaries.

- **Customer Gains:** Get to know other researchers, get knowledge, love for a research field, impact in the world, recognition as researcher.
- **Product and services:** Paper auto-formatter, Facebook of researchers, distributed journal, automated state of the art generator, universal publishing platform, Github for researchers.
- **Pain relievers:** Reviewer reputation, faster reviews, get donations for research, public and fair reviews, easier accreditation system.
- **Gain creators:** Free knowledge, open access, automatic review, find people in a research field, easier communication.

### 3.1.4 Agile methodologies

Traditional software development methodologies are being replaced by the so-called agile methodologies. These methodologies are characterized by continuous integration, iterative development and the ability to assume changes in business requirements [93; 94].

One of the most popular is known as **Extreme Programming** [95] based on a series of basic concepts when carrying out the development of a program: code simplicity and rapid prototyping, continuous customer communication with the development team, responsibility of the code of all the members of the group, short and quick meetings, refactoring and continuous integration [96].

Another well-known method within agile methodologies is **SCRUM** [97], which uses two week windows to perform development sprints and planning meetings. The use of these methodologies allow developers to create better quality software in shorter periods of time. They are designed for small teams from three to nine developers.

These two methodologies were a great influence to the methodology used to achieve the **project implementation** (see section 3.1.1). Programming smart contracts in Ethereum is a difficult task because once the source code is deployed in the blockchain, there is no way to change it. For this reason, the methodology used allowed me to develop small prototypes and run several test for each one. Every two weeks I started a development cycle consisting on programming the next prototype in the first week and running tests in the second week.

During the whole **project implementation** phase, meetings were held with the thesis directors twice a month to discuss the progress made and set short-term objectives.

As a complement to the previous methodologies, **Kanban** was used to manage the status of the project throughout the development process. Kanban was created by the company Toyota to be able to see the status of a project in an easy and fast way, but nowadays it is widely used in the field of software engineering [98]. Web services like Trello<sup>5</sup> or Github project boards<sup>6</sup> feature a digital Kanban to carry out this development process and both were used in Decentralized Science.

## 3.2 Technology

The proposed system relies upon two emerging distributed technologies. On the one hand, the Blockchain [64] provides a public decentralized ledger to record the system's interactions. On the other hand, IPFS [99] is a distributed file system to store all the papers and reviews sent to the platform. This ensures that all the information is persistent, free, accessible, and does not rely on a centralized server.

### 3.2.1 IPFS

IPFS stands for Interplanetary File System. It is a peer-to-peer file-sharing protocol that uses a cryptographic hashes to store files in a distributed network. IPFS works very similar to the HTTP protocol but in a BitTorrent way. It can be seen as a giant git repository where everyone can store, share and exchange files [100].

#### IPFS on Decentralized Science

IPFS provides a robust, distributed and secure way to store files in a decentralized network. These files are identified by a string of characters and each identifier is unique. In order to achieve this, IPFS computes the hash of the data inside the file (a cheap computational operation) and uses this hash as its identifier. It behaves like a link within the IPFS network that allows users to identify and recover the file.

This feature implies that two identical data files have the same hash, so they have the same IPFS address, eliminating duplicates in an easy, secure and fast way.

IPFS merges four main ideas: Distributed Hash Tables, BitTorrent, Git and

---

<sup>5</sup><https://trello.com/>

<sup>6</sup><https://help.github.com/articles/about-project-boards/>

Self-Certified Systems.

### Distributed Hash Tables

A distributed hash table (*DHT*) is a decentralized structure that works very similar to a hash table. It consists on a table that behaves as a collection of keys (hash strings) that identify items in a distributed database. The table performs simple mathematical operations generating a random string called *hash*. The hash acts as a pointer that directs to the table's information, allowing users to find it in a large database without performing an exhaustive search [101].

In distributed hash tables, hash strings are keys that identify a value in one or more nodes. Any node can use a key to retrieve data. This system includes a data structure called “keyspace” that is the set of all possible keys, which is broadcasted across the nodes in the system. The mapping of the keys is made by a function that calculates the *keyspace* of each node and shares it with its neighbors. These nodes also have an identifier and a set of identifiers pointing to all its neighbors nodes. If a node is removed from the network, only a small portion of the data must be recovered by other nodes [101].

This system makes *DHTs* scalable, fast and robust. It is used by frameworks such as Tapestry [102], Chord [103], Kelips [104], Kademlia [105] and IPFS [100]. These platforms are similar in cost and performance if they are tested in a large enough network. They behave very fast when it comes to searching for a key through massive networks of nodes [106], that's why it is used by IPFS to create its distributed file system.

### BitTorrent - File sharing

BitTorrent [107] is a P2P file sharing system used worldwide. In this system, files are divided into very small chunks of data, and are shared in a peer-to-peer network. Each peer aims to maximize its download rate by connecting to the best peers, meaning that peers with faster network speed will be better than ones with slow connectivity. In BitTorrent's network, peers with high upload rate will get higher download rate, so the key is balancing the network bandwidth between downloading and uploading files [108].

IPFS uses three main features from BitTorrent's protocol [100]:

- BitTorrent's data exchange protocol rewards nodes who contribute to the network, and punishes the ones who don't.

- BitTorrent tracks the availability of file chunks, sending the rarest first rather than sending the most common ones.
- IPFS uses PropShare [109], an alternative implementation of the original BitTorrent protocol designed to maximize network speed. This implementation improves the previous bandwidth allocation strategy for each peer, enhancing the download and upload speed of the network.

### Git - Version control system

A Version Control System (*VCS*) is a software to manage changes in a document, computer programs or any information. Each change is called revision and it is identified by a number, the person who did it and a timestamp. In a *VCS* revisions can be reverted to a previous version, making them useful for software development.

Git is a distributed *VCS* [110] that was born in 2005, when the development process of the Linux kernel lost its version control system. The Linux kernel is one of the biggest free software projects nowadays. It has a great team of developers behind and the code usually changes very frequently. In 2002 the team used BitKeeper<sup>7</sup> as *VCS* since they had a free license. But in 2005 when this license was over, Linus Torvalds decided to develop his own *VCS* [111].

Git was designed to be scalable and distributed, and nowadays is widely used by the open source community. The most important factors that IPFS inherits from Git are [100]:

- Git reflects changes in a file system in a distributed way using an acyclic graph, in which each revision is a node and each change is an arc.
- Objects are identified by the cryptographic hash of their contents.
- Version changes only update references and add objects. To broadcast version changes, git only needs to transfer the new objects and update the remote references.

### Self-Certifying File Systems

A self-certifying file system (*SCFS*) is a secure and decentralized file system that uses public keys to map file names, separating key management from file system security. Servers have a public key and clients use the server public key to authenticate the

---

<sup>7</sup>BitKeeper is an and distributed and scalable *VCS* available at <https://www.bitkeeper.org/>



server and establish a secure communication channel. Once the client has verified the server a secure channel is established and the actual file access takes place [112].

IPFS tries to connect these ideas into a cohesive, trustful and decentralized file system. It is build on top of a peer-to-peer network, so no nodes are privileged, and all of those store IPFS objects in local storage. These objects represent files or other data structures.

In Decentralized Science, we use this technology to store all the files in a robust and secure way, without relying on a centralized server. All these characteristics provides a persistent platform, since IPFS works on a network of thousands of nodes.

### 3.2.2 Ethereum

As seen in the section 2.2.4, Ethereum is a technology that allows its users to create fully decentralized and autonomous applications. These applications (called DApps) are smart contracts that are uploaded to the Ethereum blockchain, so it is not necessary to have a server to run or communicate with these contracts.

Ethereum is used as *backend* in Decentralized Science. All the internal operation of the platform is programmed in a smart contract and is executed from the blockchain. This implies several important features:

- **Open source:** All the source code uploaded to Ethereum is free, public and anyone can fork it in other projects.
- **Auditable interactions:** All calls and interactions to the platform are registered in the blockchain, making the entire process of scientific publication auditable.
- **Distributed platform:** All the source code is executed in the Ethereum's distributed network, meaning that Decentralized Science does not need a centralized server or a third party service to run the platform.
- **Free access to information:** Anyone can access to the platform's information (journals, authors, papers, reviews and reputation) without any cost and through a web page.

With Ethereum and IPFS, Decentralized Science is a 100 % distributed platform, open source and free<sup>8</sup> to all users.

---

<sup>8</sup>Interactions with the platform may have transaction costs within the Ethereum network (see

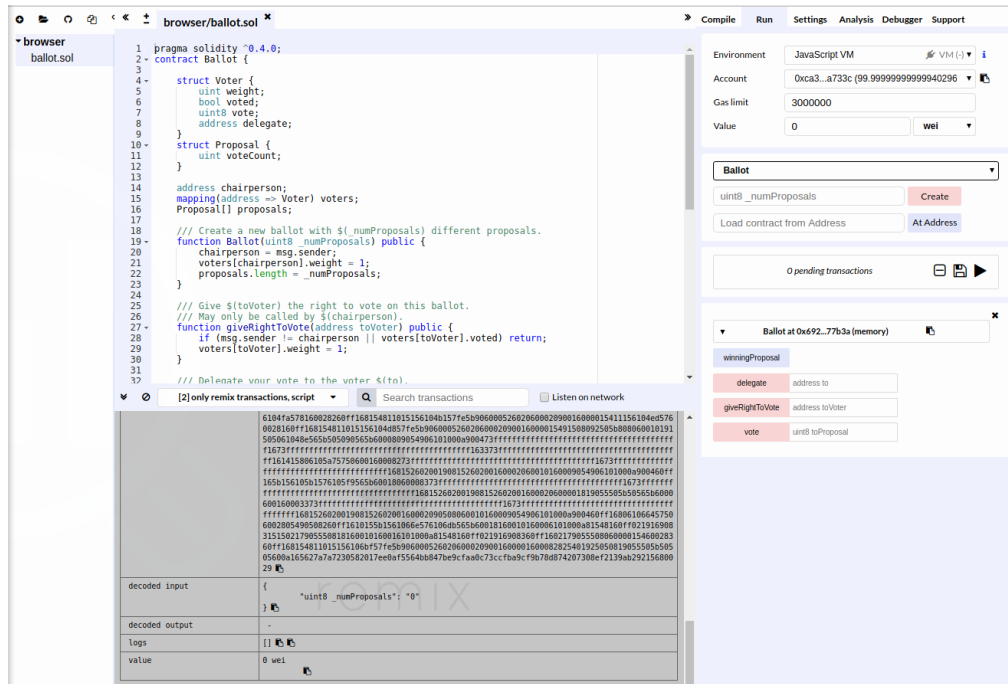


Figure 3.2: Remix smart contract web compiler.

### 3.2.3 Remix

Remix<sup>9</sup> is an online smart contract web compiler designed by the Ethereum community that allows users create, compile and deploy contracts both in a test net or in the Ethereum's blockchain.

This compiler offers the possibility to try the functionality of the contracts through a virtual machine that simulates the blockchain's behavior called EVM (*Ethereum Virtual Machine*) [113]. When deploying a contract with this platform, a series of HTML elements are generated, simulating a real deployment and allowing developers to interact with the contract, without the need of developing a front-end to test the contract behavior.

Remix also has a series of interesting features: a debugger to follow the execution process of a transaction, a gas calculator that estimates the gas cost of each transaction, a code analyzer to detect possible vulnerabilities and an ABI export system (see section 3.2.5).

section 5.6)

<sup>9</sup><https://remix.ethereum.org/>

```

testrpc -a 1000 80x24
Transaction: 0x7e00ac4f69d82e637d216f03351cb7b7049e083e06a9c7e924d245718763fd9
a
Gas usage: 0x02b5b3
Block Number: 0x09
Block Time: Mon Jan 29 2018 11:38:38 GMT+0100 (CET)

eth_sendTransaction

Transaction: 0x777fc3c281d9465a25d0fdec936e4039e32af1451e87bdade535958bb5aa09b
0
Gas usage: 0x02b5b3
Block Number: 0x0a
Block Time: Mon Jan 29 2018 11:38:38 GMT+0100 (CET)

eth_sendTransaction

Transaction: 0x9f28f45a12d4677360ed7fe8f94960003b88773e3c5d816c8b29e5fbd8f1cfc
0
Gas usage: 0x02b573
Block Number: 0x0b
Block Time: Mon Jan 29 2018 11:38:38 GMT+0100 (CET)

```

Figure 3.3: Transactions mined by testrpc in a test network.

This tool facilitates the process of developing smart contracts, since the web service allows to try different implementations of the same contract doing only small changes, without the need of installing any software (see figure 3.2).

### 3.2.4 Testrpc

Testrpc is a Node.js based Ethereum client for testing a development. It simulates an Ethereum client behavior using the `ethereum.js` library. Once installed through `npm`, it can be launched through the linux terminal with the command `testrpc` and offers the developers a series of interesting features:

- **Simulate accounts:** Testrpc<sup>10</sup> default command launches 5 accounts, but it offers the possibility to add as many as needed with the option “-accounts”, this allowed this project to simulate the interaction of 1000 accounts communicating with the contracts”. Example: `testrpc -a 1000`
- **View transactions:** Once testrpc is launched, developers can see through the console all the transactions made with the blockchain in a user-friendly way. It also has a gas<sup>11</sup> estimator, a tool that calculates the price of each transaction (see figure 3.3).
- **Create test net:** Testrpc creates a test net through the port 8545. Any user

<sup>10</sup><https://www.npmjs.com/package/ethereumjs-testrpc>

<sup>11</sup>Amount of money needed to execute a transaction (see section 2.2.4)

in the internet using Metamask (see section 3.2.5) can connect to this test net and interact with the contracts deployed.

- **Give ETH to an existing account:** If an user already has an Ethereum account and wants to connect a testrpc node, the administrator can assign ETH to that account to allow the user do transactions in the test net.

### 3.2.5 JavaScript and Metamask

In order to interact with a smart contract in Ethereum, users normally use a web application connected to the contract address (see section 2.2.4). Nevertheless, the web application must be connected to an Ethereum node in order to communicate with it. There are two different ways to achieve this. The first one is to install an specialized web browser like AlethZero<sup>12</sup>, Mist<sup>13</sup> or Brave<sup>14</sup> that provides the libraries needed to interact with smart contracts and connects to the blockchain automatically. The second one is installing an extension like Metamask<sup>15</sup> as an extension of the web browser.

Metamask is a web browser extension (available for Chrome, Firefox, and Opera) that allows its users to connect both to the Ethereum network or a custom test net. It contains the mechanisms needed to make transactions and communicate with smart contracts in the blockchain. To use it, users only need to install the extension from the Metamask homepage and configure it with an Ethereum account. Metamask also allows the creation of new accounts, but in order to make transactions with the main Ethereum blockchain, these accounts must have enough funds.

Thus, Metamask not only acts as a bridge between a web browser and the Ethereum network, it also works as a wallet and supports Ethereum's official tokens. In order to make transactions with Metamask through a web page, this web page should use JavaScript.

JavaScript is used to connect to a smart contract address and be able to call to specific functions of it. To do so, the JavaScript code must have two important data:

- **Ethereum's contrat address:** When the user loads the web page to connect to the contract, JavaScript must have the address in which it is stored, because all the interactions with the platform are made through transactions to this

---

<sup>12</sup><https://github.com/ethereum/alethzero>

<sup>13</sup><https://github.com/ethereum/mist>

<sup>14</sup><https://brave.com/>

<sup>15</sup><https://metamask.io/>

address.

- **Contract's ABI:** An ABI (Application Binary Interface) is a data structure in which JavaScript can find all the methods a contract provides. Each method defines its inputs, outputs, type, if its payable and a state. JavaScript uses this information to build the transactions to call these methods (see listing 3.1).

These two technologies were used to test the platform proposed in this work, using HTML and Testrpc to simulate a functional journal (see chapter 6)

```
{
  "constant": false ,
  "inputs": [
    {
      "name": "_reviewerAddress",
      "type": "address"
    },
    {
      "name": "_reputation",
      "type": "bool"
    }
  ],
  "name": "giveRep",
  "outputs": [],
  "payable": false ,
  "stateMutability": "nonpayable",
  "type": "function"
}
```

Listing 3.1: ABI fragment example

### 3.2.6 Github

Github<sup>16</sup> is a web hosting service for *VCS* using Git (see section 3.2.1). It offers all Git features as well as its own features like bug tracking, issues and task management, kanban boards, wiki pages among others.

Decentralized Science is 100% open software and uploaded to Github. All the source code, documents, web page, and all the information is available at:

---

<sup>16</sup><https://github.com>

`https://github.com/DecentralizedScience/`

## Part II

# Decentralized Science

This part describes the platform incrementally. First, it offers a high-level outline of the behavior designed for the platform. Next, two different implementations with technical details of each of them are described. Thereafter an analysis is made comparing the final implementation with the current systems. Finally, a conclusion is offered and several ways for future work are established.





## Chapter 4

# Platform description

*Try not. Do, or do not. There is no try.*

Yoda - The Empire Strikes Back

### 4.1 Design using a Decentralized Infrastructure

Decentralized Science is a blockchain-enabled publication system for open science. The system provides a platform for the peer review process communications, from paper submission to paper acceptance/rejection, and supports the rating of peer reviews to build a reviewer reputation network.

The proposed system relies on the technologies mentioned in section 3.2. The *Ethereum blockchain* provides a public decentralized ledger to record the system's interactions. Smart contracts are used to enforce the rules of the system, such as just accepting reviews of invited reviewers. On the other hand, *IPFS* provides a distributed content-addressable file system to store the content of the peer review process, from the first submitted paper to each of the reviews. This ensures that the information registered in the platform will be persistent, free and accessible, and will not rely on a centralized server.

The sequence diagram of the system (Figure 4.1) describes the main interactions of the platform featuring the publication process of a paper. Below we proceed to describe these interactions and the basic ideas to implement them.

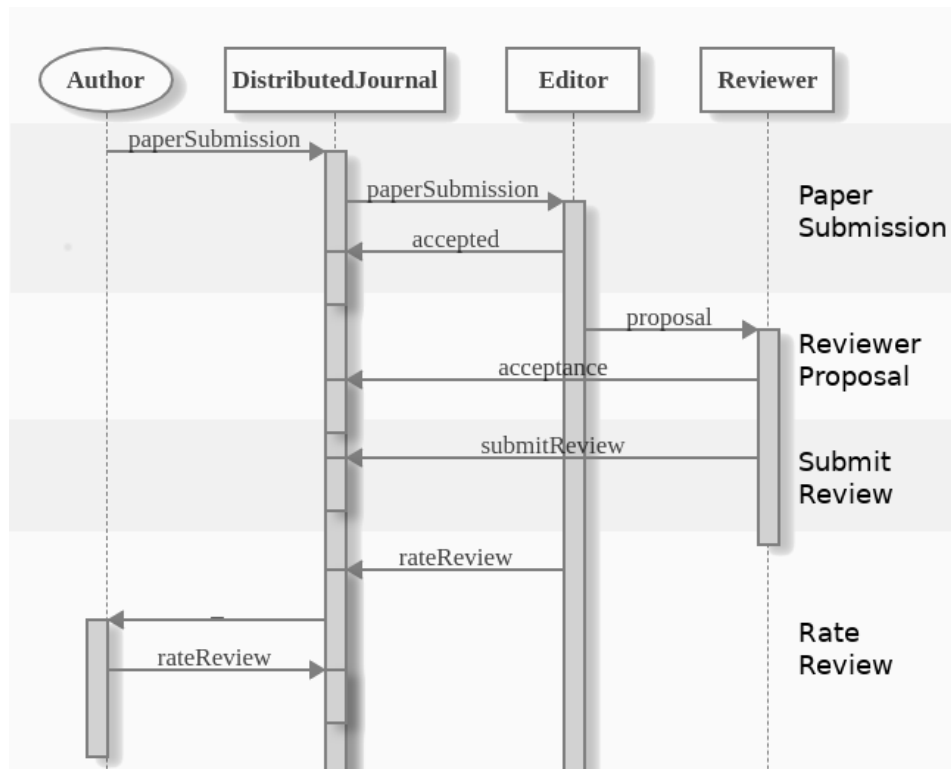


Figure 4.1: Sequence diagram of platform interaction.

### Distributed Journal

The platform is designed to fit current publication systems, offering journals to create a *Distributed Journal*, a smart contract (see section 2.2.4) used to record all interaction users make with the platform.

*Distributed Journals* have a title, a description, a list of fields of interest and an image. This information is shown in the front page of each journal and it is stored within the file system. Each *distributed journal* has an owner capable of editing the journal information, giving account privileges, assigning reviewers and changing the journal's owner.

### Account privileges

This platform allows journals to give “editor” privileges to desired users. These editors are able to assign reviewers for each paper submitted to the journal. In order to do so, editors have access to the “journal management page” in which they

can view: papers pending of reviewers assignment, papers pending of review, papers published and papers rejected.

### **Submit paper**

Users may submit a paper to a journal inside Decentralized Science. First, the platform asks for the submission information such as the authors, an abstract and the file containing the paper. Next, this information is uploaded to the file system and assigned to the journal. Once the submission is complete, the paper remains pending until an editor assigns the reviewers. At this point, the paper acts as a preprint (see section 2.1.2), remaining visible for all users until it is reviewed, modifying the paper's state from pending to published or rejected.

### **Assign review task**

Once a reviewer is proposed for a pending paper, she has the possibility to accept or reject the review task. This acceptance has a time limit in which she can communicate her decision. If the task is accepted, the reviewer has a deadline to submit the review. On the other hand, if the task is rejected, the editor can assign another reviewer.

### **Submit review**

Reviewers can submit one review for each paper they have been assigned to. This process is similar to the paper's submission as it is uploaded to the file system and then it is assigned to the corresponding paper. Each review submission has a time limit established by each journal. This time limit is notified to the reviewers before accepting the review task. In the event of a reviewer sending a review when the time has expired, a penalty is applied to the reviewers reputation inside Decentralized Science's reputation system. Finally, if all the reviews are favorable, the journal's source code automatically publishes the paper, featuring continuous publication (see section 2.1.2).

### **Rate a review**

As a novelty in Decentralized Science, there is the possibility to rate the reviews of a paper. This rating can only be done by the authors, the reviewers or the editors of the journal the paper have been submitted to. Each rating consists in an numeric

value (1 or 0) representing if the rater thinks the review is good or not. This is the first approach to the rating for the reputation system, designed to simplify the process of voting if the review is good. In the future, more complex rating systems could be explored and implemented in the platform.

When deciding whether a review is good or not there are several points of view [114; 115] and not everyone agrees. For this purpose, a community guideline is provided to explain what is considered a good review. These guidelines are created within the scientific community through academic papers and other sources, and they are always subject to change if the community requests it. In this way, this could be a solution to make all the votes in the system as fair as possible.

Subsequently, the “score” of a rating affects directly to the reviewer’s reputation in Decentralized Science. For this reason, reputation inside the system is a value between 1 and 0, meaning that reviewers with reputation closer to 1 are considered better by the community than the ones with reputation closer to 0.

## 4.2 Platform’s features

Decentralized Science consists of three main components that decentralize and try to improve three different aspects related to scientific publication:

1. Scientific papers are traditionally obtained or bought from a centralized publisher. We propose a decentralized network to distribute academic works and promote free access to science.
2. Peer reviewer quality and reliability information is difficult to predict [17], and it is usually held private by publishers and journals. The system proposes to open this information through a decentralized reputation network of peer reviewers over a blockchain.
3. Peer review governance communication is traditionally centralized and controlled by editors and publishers. Our proposal opens and decentralizes these communications making the process more transparent.

These ideas are further discussed in the following sections.

### 4.2.1 Distributed and Open Access By-Design

Open Access focuses on the free access to scientific knowledge. While publishers provide free of charge their Open Access content, their control of the science dissem-

ination infrastructure allows them to impose certain rules, such as charging authors unreasonable fees to offer their work as Open Access (Gold Open Access) [116] or the temporal embargo and restrictions on the dissemination of the final version (Green Open access) [117], among others.

This system proposes a decentralized infrastructure for science publication. Academic documents -from first drafts to final versions, including peer reviews- are shared through IPFS, an open P2P network [99] described in the previous sections. Thus, the system inherently grants Open Access by-design of its distributed infrastructure and circumvents the publishers dominant role. Moreover, the access to all those documents does not depend on the existence of this platform. Even if it ceases to exist, the documents could still be retrieved from the network.

#### 4.2.2 A Distributed Reviewer Reputation System

The information concerning the quality and reliability of reviewers is usually held private by publishers and journals (and even editors). There is no easy way to predict the quality of a reviewer from factors such as training and experience [17]. Although this information is valuable, it is kept private, reinforcing the publishers and journals influential positions.

This project extends traditional peer review communication workflow with the possibility of rating peer reviews, i.e. building a reputation system for reviewers [16]. Reviewers get rewarded for worthy, fair, and timely reviews, or penalized otherwise.

This open reputation network of reviewers could increase the visibility and recognition of the reviewing work [118; 119] as nowadays most of reviewers remain anonymous after the peer review process.

In addition, creating a public reputation network for reviewers reduces, or at least exposes, unfair and biased reviews [6; 120] (see section 2.1).

As seen in the previous section, in this approach the only ones allowed to send a rating are the authors of the paper, the editors who have assigned the reviewers, and the other reviewers of the article. Nevertheless, in future implementations, rating a review could be public, carefully considering the implications of such a decision.

There are two main implications among others:

- **The cold start problem for new papers and researchers:** Normally papers that are published in journals or conferences of low impact take a long time to gain visibility in the scientific community. However, if a reviewer reputation system is used, if three reviewers with a high reputation make a

favorable review for a paper for a little-known journal, it is likely to have more visibility since the reputation of the reviewers can positively influence the impact of the papers in the scientific community.

- **Payed reviews:** In most cases nowadays, the reviewers do not obtain any economic benefit for carrying out the revision of a paper. By implementing a reputation system, payed reviews dynamics could be formed, in which the most reputable reviewers are paid for performing reviews at the request of the journal or the authors [121]. This is an incentive for all reviewers of the system to make good reviews and gain reputation. In addition, since all interactions are in Ethereum, payments and deliveries confirmations can be made through personalized smart contracts, eliminating intermediaries, and making the process transparent and honest.

### 4.2.3 Transparent Governance

The peer review process is nowadays digitally supported, and yet some argue that the system remains feudal [11]. There are multiple proposals to improve peer review [10], and yet its communications and processes remain closed and in control of journals and publishers, and thus depend on their infrastructure [120].

The proposed system aims to improve the peer review process transparency, speed, and fairness. In order to do that, the system proposes to support the peer review interactions in an open and decentralized network. It registers, in a public decentralized ledger, the following parts of the publication process: paper submission, reviewers assignment, review submission and paper publication. Thus, processes like the selection of reviewers, or the contents of the reviews, are open to the public eye. With the interactions being time-stamped and tamper-proof thanks to the blockchain technology, they can be monitored, audited, and held accountable. Besides, more complex iterations of the system may consider blind reviews, as discussed in Section 7.1.2.

Opening the peer review process communications to the public could even change the acceptance dynamics within the system. Currently, high rejection ratios are encouraged because the risk of rejecting a relevant paper is negligible, while the acceptance of not so relevant content is penalized [11; 2]. However, within a more transparent system, the first may be penalized as well.

This transparency, combined with a distributed infrastructure for peer review, facilitates the exploration of new workflows [120].

## Chapter 5

# Architecture

*There is no secret ingredient. To make something special you just have to believe it's special.*

Mr. Ping - Kung Fu Panda

The platform architecture consists of two main parts as explained in section 3.2, a decentralized file system in which users can upload all the files using IPFS (see section 3.2.1), and a smart contract to register all interactions of the users with the platform (see section 2.2.4).

As a decentralized technology, anyone can run a node locally, connecting to the IPFS network and to the Ethereum's blockchain to interact with the platform, but this technology is not used commonly, and not all users have the knowledge to install and run these programs. As a solution, a “gateway server” to test the platform's implementation was created, using a web browser extension called Metamask<sup>1</sup> to interact with the blockchain and running an IPFS node to upload the files as explained in the diagram of the figure 5.1.

### 5.1 First Prototype

The first prototype of Decentralized Science was based on “Structs” to store the information inside the blockchain. Structs in Ethereum are similar to structs in other programming languages such as Java or C , a composite data type consisting in a group of variables placed in a block of memory. Structs in Ethereum allow

---

<sup>1</sup>See section 3.2.5

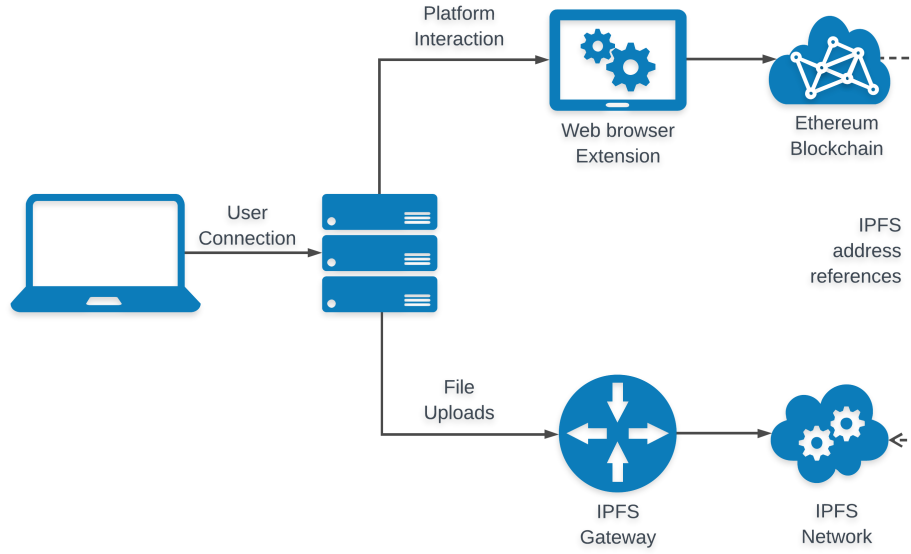


Figure 5.1: Architecture diagram of a node with IPFS and an Ethereum light client.

users to “pack” information in the contract in a solid and clear way, facilitating the code comprehensibility. In addition, storing information with this data type can suppose an improvement in the execution costs, making interactions<sup>2</sup> with the platform cheaper.

The information of this implementation is stored through an array of structs. In order to access a particular data (e.g. a paper within the platform), users only need the index within the array to retrieve it. Listing 5.1 shows the paper struct inside the *PapersLibrary* contract used in the first prototype.

<sup>2</sup>Each interaction with the platform has a small cost (see section 2.2.4)



```
contract PapersLibrary{

    ...

    struct Paper{
        // IPFS address using Multihash struct
        Multihash _ipfsPaperMultihash;

        // Ethereum's addresses of the authors
        address [] _authors;

        // Check if the paper is accepted
        bool _published;

        // Array of Reviews (see Review Struct) of this paper
        Review [] _reviews;
    }

    // Papers received are stored in this array
    Paper [] storedPapers;

    ...
}
```

Listing 5.1: Struct used for storing papers in the first prototype

## Smart Contract Architecture

Smart contracts in Ethereum can interact with each other, creating an ecosystem of programs that resemble object oriented programming. The contract structure and source code is crucial because once a contract is in the blockchain, there is no way to change it.

As explained in the section 3.1.4, one of the most important challenges in designing a contract is to reduce the cost of transactions, since the cost of these can be very high if the contract design is inefficient [122]. Developing different implementations of the same interaction in a small prototype allow developers to test each transaction in several different ways. Furthermore, inserting data in the blockchain is very expensive, so it is advisable to use the data structures that Ethereum offers to

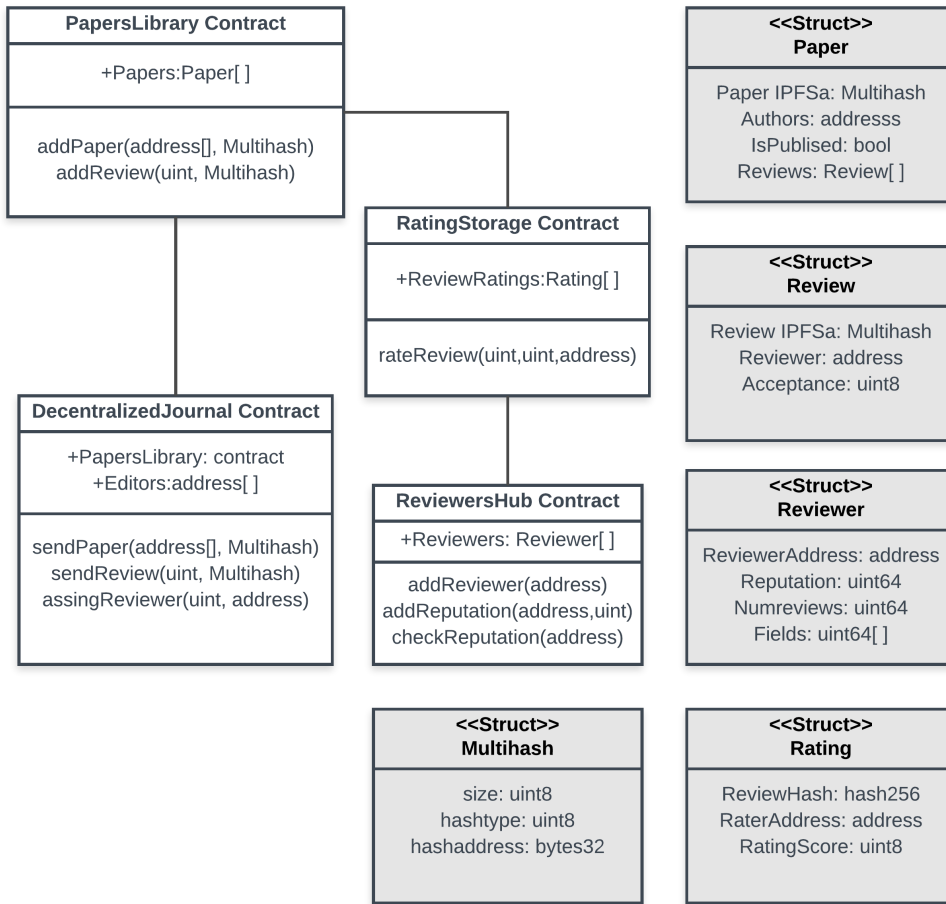


Figure 5.2: First prototype's diagram of the contracts structure.

reduce the transaction cost. Section 5.6 discusses some problems encountered during the development of the platform to reduce these costs and what solutions have been implemented.

The design of this first approach has been fragmented into 4 smart contracts that allow both to reduce the cost of transactions, and fragment the information of the platform, as seen in figure 5.2. This fragmentation allows other smart contracts interact with the information stored in Decentralized Science, without having to rely on a single contract that centralizes all the transactions. This fragmentation lets other users to “fork” part of the platform and design different implementations without altering all the system (e.g. The reputation system can be replaced without

changing the rest of the platform).

This implementation uses five different structs to store the information inside the blockchain:

- **Multihash struct:** Used to save all IPFS addresses of the platform. These addresses act as pointers inside the IPFS network to retrieve all files in the platform. It is used to store both papers and reviews in Decentralized Science.
- **Paper struct:** Papers sent to the platform are stored using this struct. It contains a Multihash as reference to the file inside IPFS network, the Ehtereum's addresses of the authors of the paper and a boolean representing if the paper is published. In addition it contains an array of the reviews of the paper.
- **Review:** This struct is used inside the Paper struct to save its reviews. In order to do so, it contains a Multihash as a reference to the review file inside IPFS, the Ethereum's address of the reviewer and an integer number between 1 and 7 showing the level of acceptance/rejection of the review meaning: strong accept (7), accept (6), weak accept (5), borderline (4), weak reject (3), reject (2), strong reject (1). As explained before, this system is part of the first approach and can be changed in the future.
- **Rating:** As part of the reputation system mentioned in chapter 4 each review can be rated to give or subtract reputation to the reviewer. This prototype uses a Rating struct to gather these and store them in the platform, containing the ratings and the users who submitted it.
- **Reviewer:** Finally the last struct used in this approach is responsible of storing the reviewers. It contains the reviewer's Ethereum address, the reputation she has, an array representing the fields of expertise and the number of reviews done by her.

Above these structures, there is a system of smart contracts which interact with each other to make the platform work. The first Decentralized Science's prototype consisted of the following contracts:

## Decentralized Journal

A Decentralized Journal (see chapter 4) is an smart contract that behaves like a digital journal. It is similar to small program in blockchain which controls all inter-

actions with authors and editors. The authors interact with this smart contract to submit papers, and the editors assign reviewers for the peer review process.

Specifically, this contract has a series of ethereum addresses associated with the editors (which are the ones who can assign reviewers) and a reference to the contract address of the “library contract”, where the papers are stored.

In addition, through this contract, the reviewers send the reviews that have been assigned to them.

## **Decentralized Library**

This smart contract stores the IPFS addresses of the papers and the reviews and controls if the papers are published.

To store this data, this contract saves a reference to the IPFS addresses. This reference is the result of the data transformation that is explained in the section 5.6, and it is used to identify a paper within Decentralized Science’s platform.

In addition, each paper stores all the revisions it has through a “Review” struct. This struct stores the review file, the address of the reviewer and an integer representing whether the paper is accepted or not.

## **Rating Storage**

This smart contract stores all the transactions about the ratings of the reviews made by the reviewers, and it is the one capable to give reputation to each one of them.

Each rating is represented by a struct that has: a hash that univocally identifies the review of a paper made by a reviewer, the address of the person making the rating and a score that represents the reputation that is given to the reviewer.

## **ReviewersHub**

It is responsible for storing the addresses, the research fields, and the reputation of the reviewers who are registered in the platform. This contract is used by: the Rating Storage contract to give the reputation scores to the reviewers, the new reviewers who want to register on the platform and the editors who want to find new reviewers to carry out the peer review process.

### First prototype problems

Using arrays and data structs implied that the data were stored in a linear way, something that is inadvisable in terms of scalability. This design could lead to a loss of performance on the platform when a critical number of papers is reached. This loss implies an increase over the transaction costs and, therefore, using the platform would be more expensive. The second prototype uses hash maps instead of arrays to store the information, a more efficient and cheaper way to store data in blockchain.

Another disadvantage of this implementation is that each of the papers was identified by the index within the library's contract array. Identifying a paper with just one number within the system did not have any advantage over the identifiers for today's digital content. The second prototype tries improve this disadvantage using Ethereum smart contract and addresses. Creating a paper smart contract in Ethereum offers several advantages: explained in the next section.

## 5.2 Second prototype

The second and final prototype of the platform is based on the use of smart contracts as papers, meaning that papers submitted to the platform have an associated Ethereum address. This implies that one of these addresses can

uniquely identify a paper inside and outside Decentralized Science. Furthermore, since all interactions with the platform are timestamped, an Ethereum address could be used to demonstrate the authorship of a scientific paper even if it has not been published.

As shown in the figure 5.3, the contract structure has been modified to eliminate the use of structs except the one used to store the IPFS addresses.

In addition, the arrays have been replaced by "mappings", a type of data in Ethereum similar to a hashmap that improves the costs of read and write operations within smart contracts.

The smart contracts used in the second prototype are the following:

### Journal contract

This smart contract is similar to the first prototype. Its function is to manage the publication process. It has an owner who controls the journal and a series of editors who are responsible for assigning the reviewers of the articles.

The main difference with the previous version is that every time someone sends

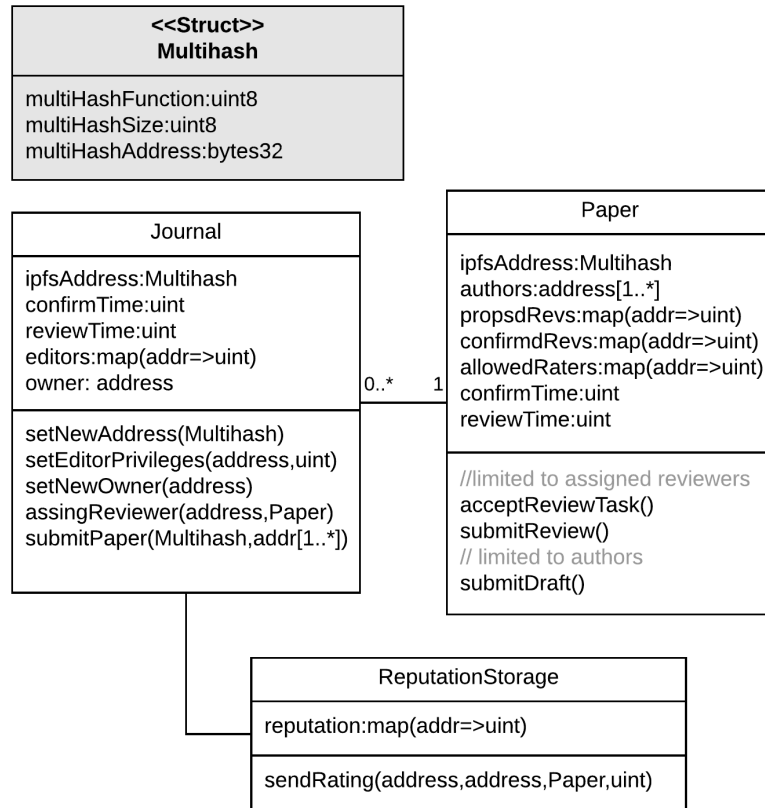


Figure 5.3: Second prototype's diagram of the contracts structure.

a paper to the journal, it creates a paper smart contract containing the submission information.

Journal's owners can set time limits for the confirmation time (explained in chapter 4) and for the review time. This information is public, meaning that researchers can decide if they want to submit a paper to a certain journal depending on the review times they are looking for.

Finally, this contract has the possibility of adding information such as a title, a short description, an image, and a list of topics in which the journal focuses.

### **Paper contract**

The biggest contribution of this prototype is that the papers are Ethereum contracts, therefore they have an associated address (see section 2.2.4). Since all the addresses in Ethereum are unique, they could identify a paper both inside and outside the platform.

The paper contract keeps a reference of the journal to which it has been submitted. This allows the journal's editors to assign the reviewers for the peer-review process.

Once the reviewers of a paper have been assigned, they only have to interact with the paper contract without having to go through the journal, thus freeing the transaction traffic towards the same address. When the reviews have been made and if they are favorable, the paper is published automatically.

Besides, each time a review is received, the paper contract allows a rating to be submitted. In this way, it is the paper itself that controls who can perform the rating and who has done it already.

The paper contract also offers the possibility to receive money transactions, as a smart contract can transfer and store Ether<sup>3</sup>. This feature could open new dynamics of research funding, where researchers receive money for their projects through academic papers within the platform.

Finally, the papers contract addresses can suppose a new citation system for the academic process, in which the papers have links to the paper addresses they cite.

### **RepHub contract**

This contract is the reputation system, responsible for storing the reputation of each reviewer. For this, whenever someone wants to make a rating, this contract communicates with the corresponding paper to verify that the person who is sending the rating can do so.

If the paper verifies that it can be done, an internal transaction is sent between the reputation system and the paper to indicate that the person has made a rating. Once this is done, this contract calculates the reviewer's reputation and stores it within the system.

The platform uses these contracts to perform all design interactions. Section 5.5 explores more in depth each of these with activity diagrams.

---

<sup>3</sup>Ether is Ethereum's currency (see section 2.2.4)

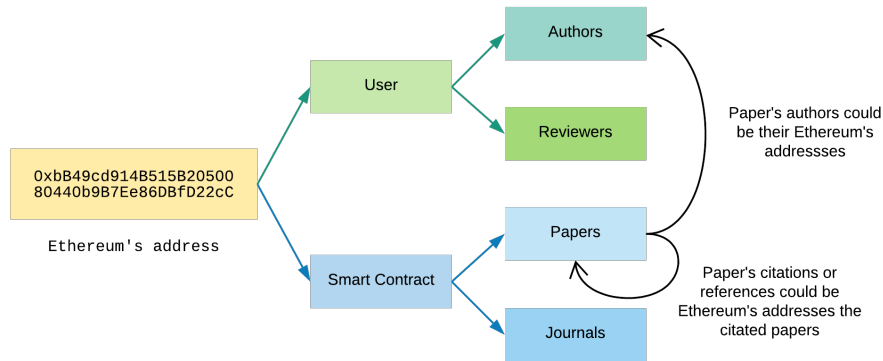


Figure 5.4: Use of Ethereum's addresses in Decentralized Science.

### 5.3 Ethereum addresses in Decentralized Science

Ethereum addresses (see section 2.2.4) are strings of characters that identify an individual or a contracts in the blockchain. These addresses are unique, meaning that two people or contract cannot have the same address. As mentioned in this chapter, inside Decentralized Science users are authors, reviewers, editors or some of these roles simultaneously. This implies that an ethereum address can be used to identify and verify an individual's scientific career if there are enough interactions with the platform. Each of these is registered in the blockchain as a transaction that have a transaction id, which can be used to obtain the information of that particular transaction as shown in the figure 5.4. This could mean that in the future, when researchers need to give credit of their careers, they would only need to supply their ethereum's addresses (see Figure 5.5).

Ethereum's smart contracts also have an ethereum address, meaning that all papers and journals in the platform have its own Ethereum address. Although this address is unique, it can be used to identify a paper or a journal outside the blockchain (e.g. with a QR code). This system could potentially replace digital identifiers like ISBN or DOI.

### 5.4 Reputation system

As mentioned in the chapter 4, paper reviews can be rated by the authors of the paper, the editors who have assigned the reviewers, and the other reviewers of the



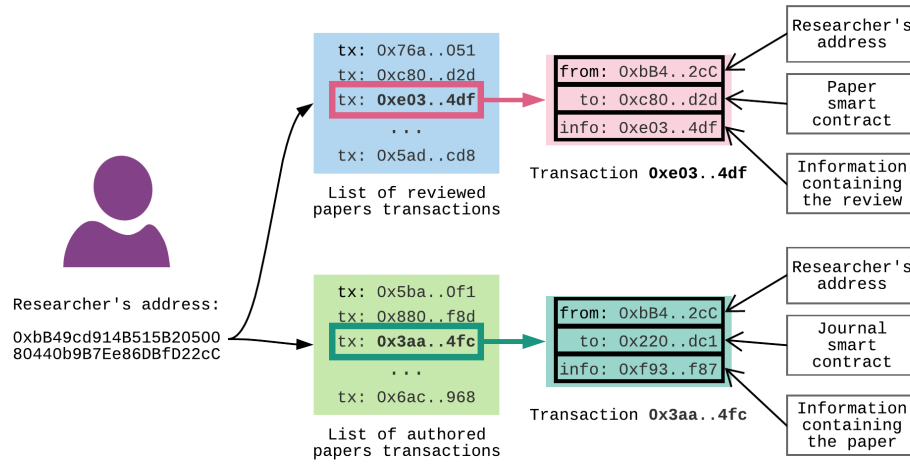


Figure 5.5: Researcher's transactions list.

article. This first approach attempts to avoid direct attacks on the reputation of its users, which limits the possibility of rating a review to the authors and other reviewers of a paper.

On the other hand, if a review is unfavorable, there is a possibility that the authors will misqualify the reviewer, submitting a bad rating. For that reason, all the ratings are public and visible by anyone, trying to discourage this type of behavior.

To go deeper into the subject, the internal operation of the system follows the following steps:

1. When a reviewer submits a review, the paper's smart contract allows the authors and the other reviewers to submit one rating for that particular review.
2. In order to send a rating, users should make a transaction to the RepHub smart contract with the score, the paper's address and the reviewer's address. These two data univocally identify a review, since the same reviewer can not make two different reviews of the same paper.
3. For each rating, the system registers the rater and modifies the reviewer's reputation, making an exponential smoothing of the score received [123]. In this case, exponential smoothing is used to calculate the average of the score

without knowing the total number of raters. To be more precise, the alpha value used in the exponential smoothing is 0.2.

Within a reputation system there will always be some controversy and there will always be methods to placate it [124], but the design presented in this work is a proof of concept that will evolve as new systems are designed to appease the problems that have arisen.

## 5.5 Platform interactions and activity diagrams

In the following section we will discuss the activity diagrams for each transaction users can make to interact with the platform.

Each of these interactions must be done through an Ethereum personal address, as it works as an identifier of the user performing it. For this reason, all of the following activity diagrams ask the user for an Ethereum address. In case a new user uses the platform does not have a personal address, the platform has the possibility to create one for that user through the Metamask<sup>4</sup> extension. Furthermore, in order to complete each transaction, an essential requirement is that the user must have an Ethereum account and Metamask installed.

### Send paper

Once an user has an Ethereum account she may send papers. The platform asks for the authors and for the file through a web form. Javascript collects this information and uploads it to the file system, generating an IPFS file address. Finally, the platform creates a transaction with this data and sends it to the platform. See Table 5.1 and Figure 5.6.

### Assign Reviewer

After receiving a paper, an editor has the possibility to assign reviewers. To do so, there must be a journal in the platform and the user must have editor privileges. Next, the editor assigns the reviewers (with which she has previously contacted) of the paper and the platform creates the review tasks for each reviewer. These tasks can be accepted or rejected by the reviewers and they can have a “deadline” to receive the acceptance. See Table 5.2 and Figure 5.7.

---

<sup>4</sup>Used to connect a web browser to Ethereum (see section 3.2.5)

AD1	Send Paper
Description	The user uploads a file to IPFS and creates an Ethereum transaction
Input	Authors' addresses and a file
Output	IPFS address of the paper
Requirements	Authors' Ethereum addresses and Metamask installed
Precondition	The journal's contract address exists and the IPFS node is online
Post-condition	There is a transaction to the journal's contract address with the information about the authors and the IPFS address of the paper

Table 5.1: Requirements of sending a paper.

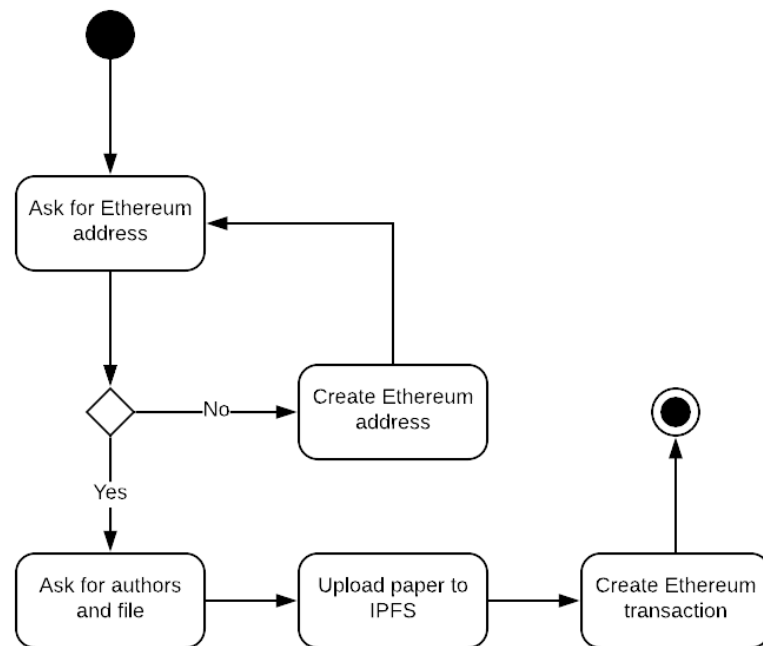


Figure 5.6: Activity diagram of sending a paper.

AD2	Assign Reviewer
Description	An editor assigns reviewers for a paper
Input	Ethereum's addresses of the reviewers
Output	
Requirements	The journal's contract address exists
Precondition	The address has editor permissions and there is a paper pending
Post-condition	A transaction is created to assign the reviewers

Table 5.2: Requirements of assigning a reviewer.

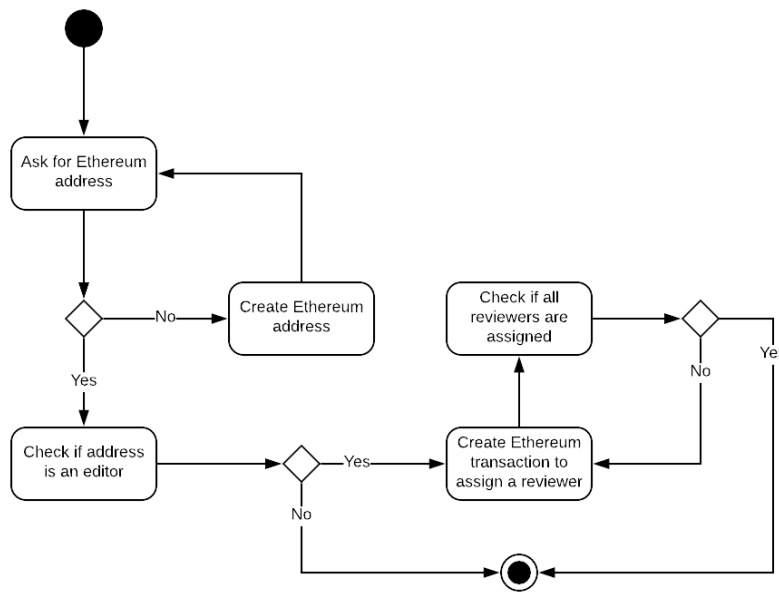


Figure 5.7: Activity diagram of assigning a reviewer.

### Accept review task

As explained before a reviewer has the possibility to accept or reject a review task. Before doing so the reviewer must have an Ethereum address. Then, when a task is received, the reviewer has a time limit to accept it. If it is not accepted, the editor can assign another reviewer. In the other hand, if it is accepted, a transaction is created and the reviewer is registered in the paper contract to be able to send his review. See Table 5.3 and Figure 5.8.

AD3	Accept Review task
Description	An address authorized as a reviewer accepts the review task
Input	A boolean accepting the task
Output	
Requirements	The journal's contract address exists
Precondition	There are a petition pending for the reviewers address
Post-condition	The petition is accepted and there is a deadline to submit the review

Table 5.3: Requirements of accepting a review.

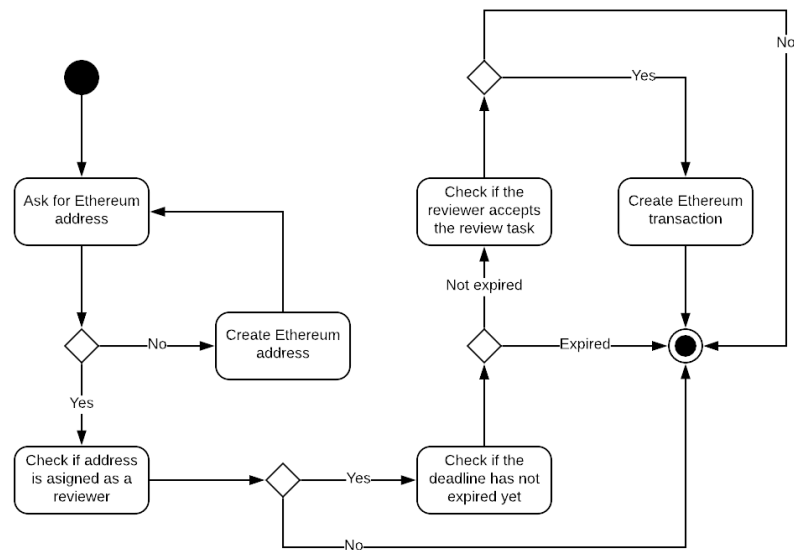


Figure 5.8: Activity diagram of accepting a review.

## Send review

Sending a review is similar to sending a paper. The main difference is that the reviewer, once she accepts the review task, has a deadline to submit a review. If the review is sent before the time limit, the file containing the review is uploaded to IPFS and the platform creates an Ethereum transaction with this information and the acceptance level (see section 4.1). Otherwise, if the review is not sent, the reviewer loses one point of reputation within the reputation system. See Table 5.4 and Figure 5.9.

AD4	Send Review
Description	A reviewers submits the review previously accepted via IPFS
Input	A file containing the review
Output	The IPFS address of the review
Requirements	The journal's contract address exists
Precondition	The submission is within the deadline
Post-condition	There is a transaction to the journal's contract address

Table 5.4: Requirements of sending a review.

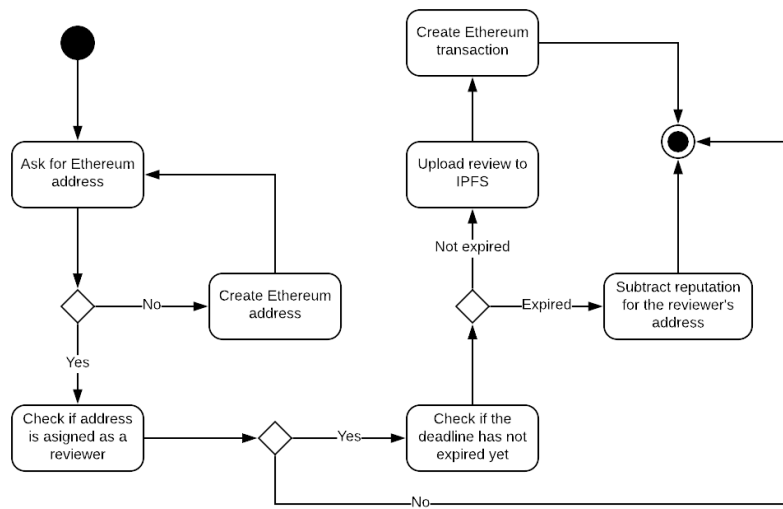


Figure 5.9: Activity diagram of sending a review.

### Rate review

As part of the reputation system, all reviews can be rated by the authors and by other reviewers. When a review is sent to the platform, it generates an identifier which can be rated with 1 or 0 as explained in section 5.4. See Table 5.5 and Figure 5.10.

AD5	Rate Review
Description	The authorized address submits a rating about a review
Input	An integer from 0 to 1
Output	
Requirements	The journal's contract address exists
Precondition	The address can submit a review and has not done it yet
Post-condition	The reputation of the reviewer is affected by de rating

Table 5.5: Requirements of rating a review.

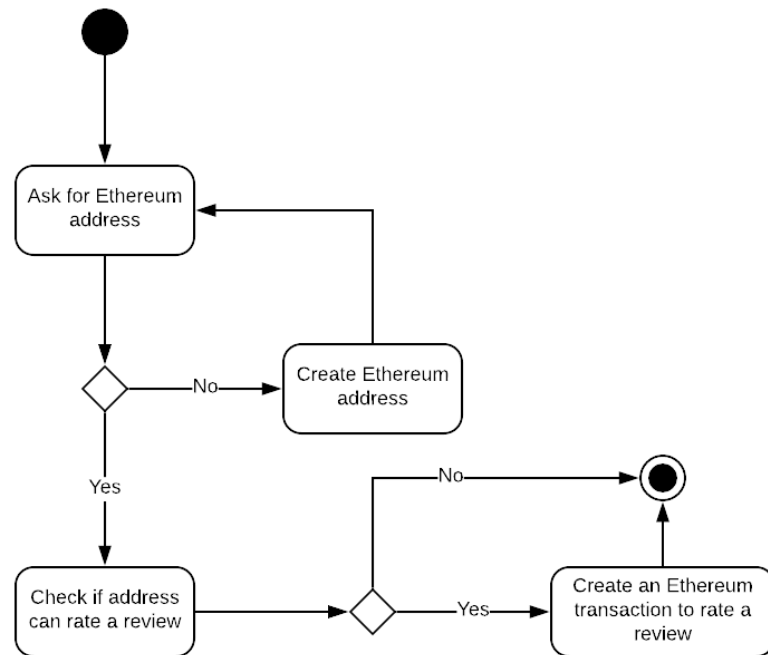


Figure 5.10: Activity diagram of rating a review.

## 5.6 Transaction costs

As explained in the section 3.2.2 Ethereum's smart contracts are transaction-based. Each interaction with the platform is registered in the blockchain with a transaction. The architecture of this platform allows the following transactions (all the costs are estimated<sup>5</sup>).

- **Create the platform:** To create Decentralized Science, there must be an initial transaction with the source code to upload it to the blockchain. Normally this transaction is the most expensive, and is only necessary to do it once.  
  
This transaction creates all the contracts required by the platform and outputs the Ethereum's address of the platform. The gas amount required to this transaction is 1.696.291, approximately from 2 to 4 euros.
- **Send Paper:** To send a paper the user have to send a transaction containing the Ethereum addresses of the authors and a file containing the paper. This file will be uploaded to IPFS through a node and then the resulting address will be inserted in the transaction. The gas amount required to do this transaction is 114.812, around 0,30 euros.
- **Assign reviewers:** Assigning the reviewers is normally very cheap because an address authorized as "editor" must create a simple transaction with the paper identifier and the addresses of the reviewers. The gas amount required to do this transaction is 58.707, around 0,10 euros.
- **Accept review:** This is the cheapest interaction with the platform, a reviewer must accept the task of reviewing a paper through a transaction. With this system, the reviewer cryptographically signs the contract and compromises to review the paper. The gas amount required to do this transaction is 23.971, around 0,04 euros.
- **Send Review:** Sending a review is also expensive, because the review is also a file and must be uploaded to IPFS. The transaction also contains a reference to the paper, and an integer representing the acceptance level of the review. The gas amount required to do this transaction is 149.760, around 0,35 euros.
- **Send Rating:** This transaction contains a reference to the review of a paper, and an integer from 1 to 5 representing the rating of a review. This transaction

---

<sup>5</sup>Check <https://ethgasstation.info/> for more info about transaction costs



initially can only be made by the other reviewers, the authors and the editors of the journal. The gas amount required to do this transaction is 94.122, around 0,04 euros.

### 5.6.1 Reducing transaction costs

In Ethereum, reducing the transaction costs is critical, because all interactions have costs to the user. In the Decentralized Science's contract the following actions have been taken to reduce these costs:

- **Reduce the amount of data stored in the blockchain:** Storing data in the blockchain is slow and expensive, so the platform only registers the minimum amount of data necessary to verify the interactions between users. All data files are stored in IPFS and the addresses are saved in Ethereum.
- **Avoid expensive data types:** Working with “Strings” in *Solidity* is very expensive, that's why it is recommended to avoid using this data type.

As described in the chapter 4. Decentralized Science stores IPFS addresses as links between Ethereum and IPFS, but these addresses are *Base58* strings, a data type not implemented yet in Ethereum. This address has 3 parts: 1) A number representing a hash function, 2) another number representing the size of the file and 3) a hashed string of the data done with the 1) hash function.

To save an IPFS address in Ethereum, initially a “String” data type was used. But to avoid using this type, first the *Base58* is decoded into the three parts mentioned above. Finally instead of creating a transaction with a string representing the IPFS address, the transaction contains the hash ID (using an *uint8* data type), the size (using an *uint8* data type) and the hashed data (using *bytes32* data type). This allows the platform to reduce the transactions involving a file upload around 40%.



## Chapter 6

# Proof of concept

*Sooner or later you're going to realize just as I did that  
there's a difference between knowing the path and  
walking the path.*

Morpheus - The Matrix

### 6.1 Platform showcase

As proof of concept, this work provides several files to run and test a local server with the prototype of Decentralized Science. The server consists of a web interface that allows users to access to the functionality of the distributed system (IPFS + Ethereum) without the need to know these technologies.

Users can interact with this platform and connect to a blockchain test net in which all the accounts have unlimited money to make all the transactions to interact with Ethereum.

This prototype is a decentralized system, users may run a local node instead of connecting to the gateway provided. The source code of the platform is open source, and can be obtained and deployed in an Ethereum and IPFS node. Figure 6.1 shows a mockup to illustrate the platform's structure and rotation among its contents.

It should be noted that this web framework acts as a layer between blockchain and user interaction and it can be modified by any user since the internal design of the platform is inside blockchain.

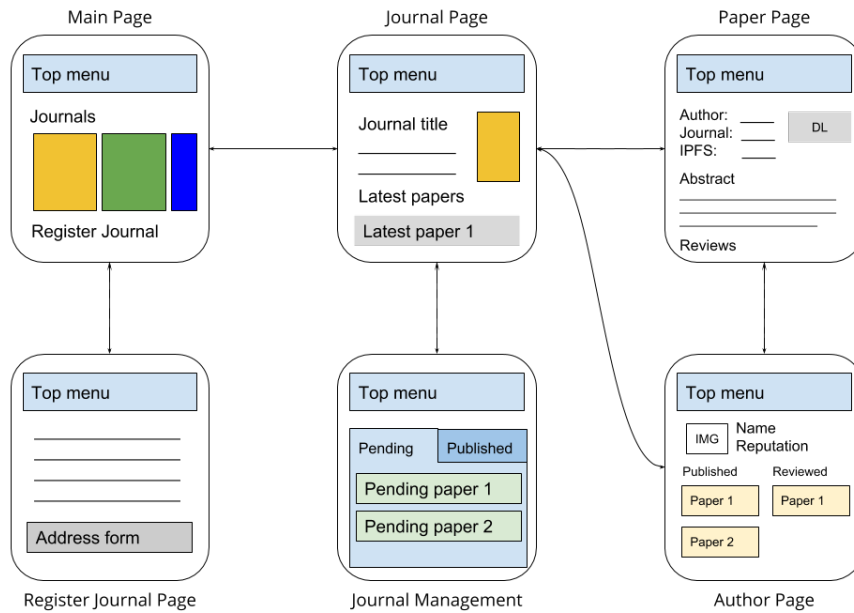


Figure 6.1: Mock-up of the possible interactions of the platform's front-end.

## Front-page

As showed in the figure 6.2, the front page of the platform shows three main sections:

- **A showcase showing the journals registered:** All journals registered in the platform are automatically showed in the front page in order of popularity. Showing the most popular journals first (i.e. journals with the most publications).
- **Register journal link:** A link to the journal registration page.

The platform also has a searchbox to search queries within the platform.

## Register Journal

Decentralized Science offers the possibility to register journals within the platform. Journal editors who want to adopt this system, can implement a smart contract template hosted in github, upload it to Ethereum and provide the address the registration page. This will create a transaction in Ethereum and will link the new registered journal in the platform.

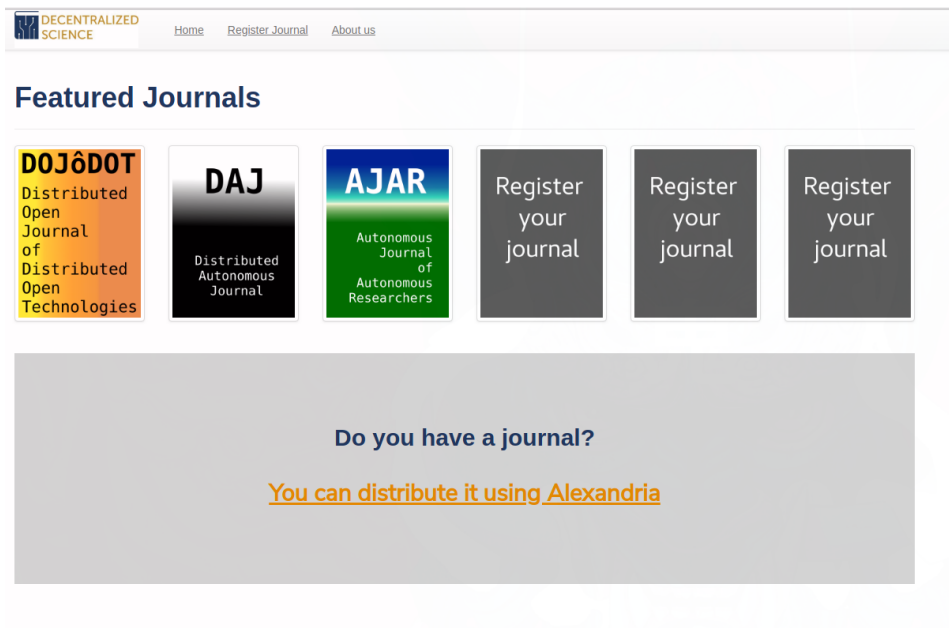


Figure 6.2: The front page of the platform.

## Register your Journal

Text Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

### Get the template journal

You can get the template of the Decentralized Autonomous Journals [HERE](#).

### Register your Contract

Ethereum Contract address:

Figure 6.3: Register journal page.

The figure 6.3 shows the registration page for journals. The HTML form creates an Ethereum transaction to Decentralized Science's address. The address provided must fit decentralized journal template, otherwise the transaction will be unsuccessful. In this prototype the journal smart contract programming and deployment should be done outside this platform following the template provided, but in future implementations this feature will be inside the framework.

## Journal Page

The journal page contains all the information about the journal:

- Title of the journal.
- A short description about the journal.
- An image of the journals cover.
- Latest papers published and submitted (preprints) with a corresponding link to IPDF.
- A list to all the papers of the journal.
- A link to the paper submission page (see figure 6.5).
- The average review time of the papers published in the journal.

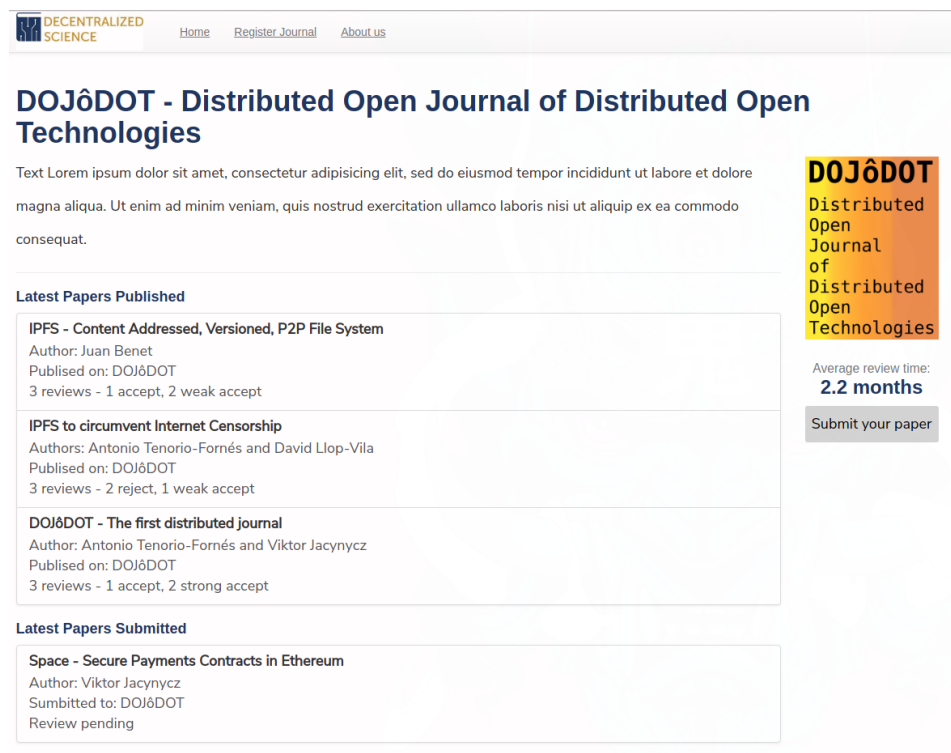


Figure 6.4: Journal page.

### DOJôDOT - Submit a paper

**Title**

Title

**Abstract**

**Authors**

Author Address

**File input**

Choose File No file chosen Add Author

**Topics (From ACM Computer Classification System)**

Submit

[Go back](#)

Figure 6.5: Submission page.

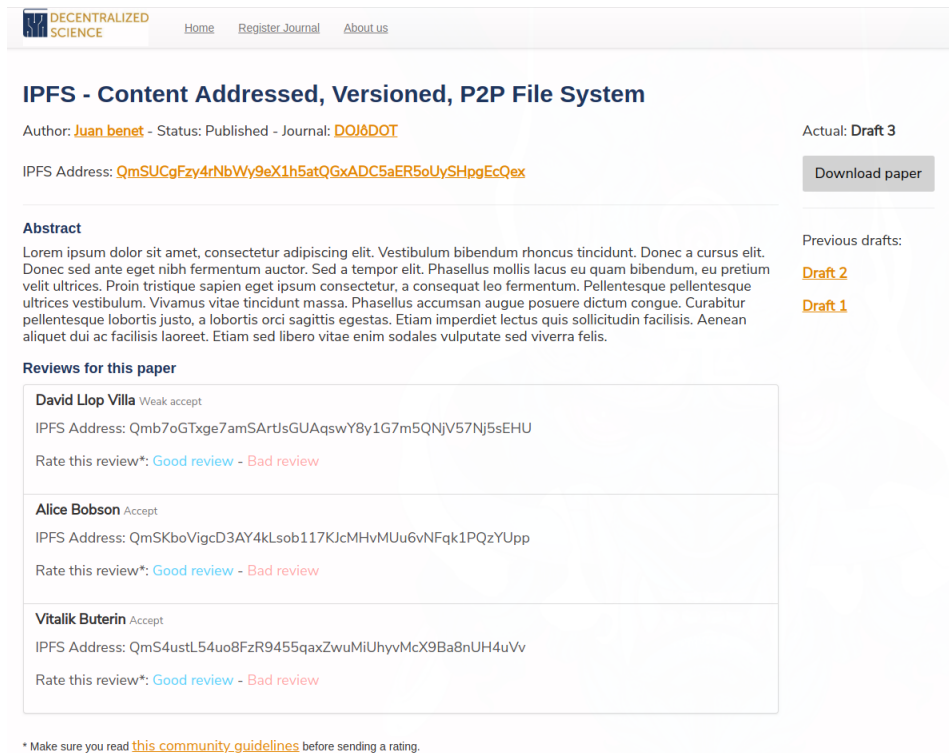
The figure 6.4 shows the journal's page with all the information mentioned above. The HTML connects to the journal's Ethereum address to get the IPFS address of the journal. This last address references a *JSON* file containing all the information about the journal.

### Paper page

A paper page contains useful information about the paper:

- Title.
- Authors, with links to each author page.
- Abstract of the paper.
- A download link for the paper and links to download each draft, offering the possibility to view the previous work of the researchers.
- The reviews from the reviewers showing the author, the IPFS link with the text of the review and the acceptance level. Each review can also be rated if the user's address can send the rating.

The figure 6.6 shows the paper's page with all the information mentioned above.



**DECENTRALIZED SCIENCE** Home Register Journal About us

## IPFS - Content Addressed, Versioned, P2P File System

Author: [Juan benet](#) - Status: Published - Journal: [DOJ&DOT](#)

Actual: **Draft 3**

IPFS Address: [QmSUCgFzy4rNbWy9eX1h5atQGxADC5aER5oUySHpgEcQex](#)

**Download paper**

**Abstract**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum bibendum rhoncus tincidunt. Donec a cursus elit. Donec sed ante eget nibh fermentum auctor. Sed a tempor elit. Phasellus mollis lacus eu quam bibendum, eu pretium velit ultrices. Proin tristique sapien eget ipsum consectetur, a consequat leo fermentum. Pellentesque pellentesque ultrices vestibulum. Vivamus vitae tincidunt massa. Phasellus accumsan augue posuere dictum congue. Curabitur pellentesque lobortis justo, a lobortis orci sagittis egestas. Etiam imperdiet lectus quis sollicitudin facilisis. Aenean aliquet dui ac facilisis laoreet. Etiam sed libero vitae enim sodales vulputate sed viverra felis.

**Previous drafts:**

[Draft 2](#)

[Draft 1](#)

**Reviews for this paper**

<p><b>David Llop Villa</b> Weak accept</p> <p>IPFS Address: <a href="#">Qmb7oGTxge7amSArUsGUAqswY8y1G7m5QNjV57Nj5sEHU</a></p> <p>Rate this review*: <a href="#">Good review</a> - <a href="#">Bad review</a></p>
<p><b>Alice Bobson</b> Accept</p> <p>IPFS Address: <a href="#">QmSKboVigcD3AY4kLsob117KJcMHvMUu6vNFqk1PQzYUpp</a></p> <p>Rate this review*: <a href="#">Good review</a> - <a href="#">Bad review</a></p>
<p><b>Vitalik Buterin</b> Accept</p> <p>IPFS Address: <a href="#">QmS4ustL54uo8FzR9455qaxZwuMiUhyvMcX9Ba8nUH4uVv</a></p> <p>Rate this review*: <a href="#">Good review</a> - <a href="#">Bad review</a></p>

\* Make sure you read [this community guidelines](#) before sending a rating.

Figure 6.6: Paper page.

### Author's page

Each time a new author submits a paper to a journal its address is registered. The authors page shows all the information about a researcher, connecting to the journal contract and the reputation contract. The main features about this page are:

- **Reputation:** The reputation of a researcher is represented with a 5-star rating, depending on how good or bad is as a reviewer. This star rating is calculated by the HTML depending on the reputation score (see section 4.2.2), transforming 0 to 1 score into 0 to 5 stars. It also contains the percent of the reviews delivered in time.
- **Ethereum's address:** The address of the researcher can be used to verify all the transactions done with that address. This offers the possibility to check a researcher career with only its address.
- **Papers published:** All the papers published as a researcher with the men-



tioned address.

- **Papers reviewed:** All the reviews the researcher has done.



The screenshot shows the author's profile for Viktor Jacynycz on the Decentralized Science platform. The header includes the platform logo and navigation links. The profile section displays a profile picture, the author's name, their Ethereum address, the journal they published on (DOJ&DOT), and their rating (5 stars) and score (1). Below this, there are two columns: 'Authored Papers' and 'Reviews on other papers'. The 'Authored Papers' column lists two papers: 'DOJ&DOT - The first distributed journal' and 'Space - Secure Payments Contracts in Ethereum'. The 'Reviews on other papers' column lists one paper: 'IPFS - Content Addressed, Versioned, P2P File System'.

Authored Papers	Reviews on other papers
<b>DOJ&amp;DOT - The first distributed journal</b> Author: Antonio Tenorio-Fornés and Viktor Jacynycz Published on: DOJ&DOT 3 reviews - 1 accept, 2 strong accept	<b>IPFS - Content Addressed, Versioned, P2P File System</b> Author: Juan Benet Published on: DOJ&DOT 3 reviews - 1 accept, 2 weak accept
<b>Space - Secure Payments Contracts in Ethereum</b> Author: Viktor Jacynycz Submitted to: DOJ&DOT Review pending	

Figure 6.7: Author's page.

The figure 6.7 shows the author's page with all the information mentioned above.

### Journal's management page

The platform allows addresses with editor privileges to assign reviewers to pending papers. The page shows the pending, accepted and rejected papers as shown in the figure 6.8.

For each pending paper there is a modal window to assign the addresses of the reviewers as shown in figure 6.9. This will create an Ethereum transaction to make the assignment public, and allows the reviewers assigned to accept the review.

## DOJôDOT - Manage Journal

[Go Back](#)

<a href="#">Pending Papers</a>	<a href="#">Accepted Papers</a>	<a href="#">Rejected Papers</a>
--------------------------------	---------------------------------	---------------------------------

**Personal coins - Distributed debit system**  
Author: David Llop-Vila  
Published on: DOJôDOT

No reviewers assigned

**Space - Secure Payments Contract in Ethereum**  
Author: Viktor Jacynycz  
Published on: DOJôDOT

2 of 3 reviewers assigned

Figure 6.8: Journal's management page.

### Personal coins - Distributed debit system

×

#### Assigned reviewers

No reviewers assigned

#### Assing reviewer

Assign Reviewer

Close

Save changes

Figure 6.9: Assign modal window from journal's management page.

# Chapter 7

## Discussion

*I am looking for someone to share in an adventure that  
I am arranging, and it's very difficult to find anyone.*

The Hobbit - J. R. R. Tolkien

As a result of the development of this platform, the ideal scenario would be that journals with impact factors like the mentioned *JCR* adopt this system. There are some important points to consider if this platform becomes widely used.

### 7.1 Issues and possible solutions

All interactions with the platform are registered in a chain of blocks, which means they are public. This could be a major problem when it comes to anonymity in the peer review process.

Anonymity of reviewers and authors in peer reviews is traditionally used to improve the fairness of the process. Thanks to single blind reviews, anonymous reviewers can honestly criticize a paper without fearing the reactions of the authors. Double blind reviews also allow to reduce the impact of personal biases. Finally, open review models propose that both authors and reviewers know each other. Nevertheless, the anonymity of the reviewers can also be abused. Unfair or low quality reviews were not discouraged by the system due to the lack of consequences.

#### 7.1.1 Privacy for the peer review process

As mentioned in previous sections, all the information is stored in either IPFS (files) or Ethereum (interactions). All this information is public for everyone, which can be

an important privacy problem for the reviewers if they do not want to reveal their identity. Exploring different possible configurations for each model of peer review may solve this problem.

Each of the anonymity options of the system requires different solutions, which are discussed below. The question of whether we can keep the benefits of blind review while providing accountability and recognition to reviews deserves special consideration.

### **Blind peer review**

Blind review is the protection of the identity of reviewers in the peer review process. This way, anonymous reviewers can honestly criticize a paper without fearing the reactions of the authors. In a blockchain, this protection could be easily achieved by using single-use addresses previously agreed with the editor.

### **Double blinded peer review**

A double blinded review is a blind review that additionally protects the authors identity to prevent social bias [125; 126]. Authors could protect their identities prior to publication by providing a single-use public address on submission. Later they can reveal their real identity since they are the only ones with access to that address.

### **Open peer review**

Open review models propose that both authors and reviewers know each other [19]. While studies found effect on the percentage of reviewers declining to review [127] other implications remain open to debate [128].

## **7.1.2 Privacy for the rating process**

This section builds on the previous section, adding a new layer of complexity: we do not only deal with reviews, but with both reviews and ratings. As already proposed in Section 5.4, the construction of a reputation network of reviewers may improve the accountability of the peer review process. Thus, this section explores the different privacy settings such reputation systems may have. One of these settings, the rating of blind reviews, is explored in more detail. Challenges of such system are identified, and will later guide Section 7.1.3 discussion on how it may be achieved.

	<b>Public Reviewer</b>	<b>Anonymous Reviewer</b>
<b>Public Rater</b>	Signed rate of open review	Signed rate of blind review
<b>Anonymous Rater</b>	Anonymous rate of open review	Anonymous rate of blind review

Table 7.1: Different configurations to rate a review

### Signed Rating

Similarly to the open peer review (explained in the previous section), signed ratings are public and verified ratings of a review. It is straight forward to implement by maintaining a public identity for the raters.

### Anonymous Rating

Protecting the identity of raters is interesting in several reputation systems [129]. We can support this anonymity feature using *blinded tokens* [129] that grant permission to rate without revealing the identity of the rater. People authorized to rate a review in the system, e.g. authors, editors and other reviewers of the paper involved in the process, may each get one of these tokens.

### Rating Blind Reviews

The question of whether we can keep the benefits of blind review while providing accountability and recognition to reviewers (and thus rating their reviews) deserves special consideration.

The following challenges must be considered in order to provide this privacy setting:

1. **Anonymity:** The reviewer should be able to claim the rating received in her review (e.g. to receive positive reputation) without revealing that she is the author of the review.
2. **Accountability:** The reviewer should not be able to avoid the effect of negative reviews (e.g. claiming just the positive ratings).
3. **Authorization:** The ratings should come from authorized raters (i.e. mini-

mizing cheating).

4. **Sybil resistance:** Having several identities in the system should not provide advantages. Note blockchain systems such as Ethereum allow the creation of multiple identities per user.
5. **System abuses:** The anonymity of the interactions may hinder the detection and prevention of system abuses. For instance, malicious actors may try to submit fake reviews to be rated by accounts they control in order to get unfair good ratings. Detecting this behavior would not be trivial since reviews and ratings may be anonymous.

A system allowing an anonymous, yet accountable, reputation system for peer reviewing would enable a new privacy and accountability model for peer review. However, its implementation face important challenges such as those described above. Next section provides an overview of how existing techniques may be applied to tackle the identified challenges.

### 7.1.3 Achieving Accountable Anonymous Reviews

The previous section identifies challenges that an anonymous yet accountable reputation system for peer reviews face. Some existing technologies have been applied to similar challenges, and others may help to combine their advantages. This section explains these technologies and how they may be used to tackle the challenges of this system. First it provides an overview of how the technologies may be combined, and afterwards a description of the technologies follows.

A simple way of protecting the identity of users is the use of different virtual identities for each interaction, i.e. *single-use identities*. However, linking the reputation received by this single-use identities to their real identity both providing accountability (Challenge 2) and preserving the anonymity (Challenge 1) require the use of other technologies.

In order to provide accountability (Challenge 2), the system may try to detect when an identity has not claimed back a bad reputation. For this purpose, a reputation deposit or *collateral* could be requested for each rating a reviewer may get. This way, users could compare the number of claimed ratings and the number of unclaimed ratings, and assume bad ratings for those that are missing. This collateral-based technique should be applied carefully, avoiding abuses such as trying to use the same collateral for different ratings. Advanced cryptographic techniques

such as *zk-SNARKs* (explained below) may help to prove that these requirements are met without compromising the reviewers' identity. These techniques may be used to allow a reviewer to claim a rating from a review she did without revealing her identity but proving her authorship (Challenge 1).

A different issue is to allow ratings to come solely from authorized raters (Challenge 3). To attend these authorization requirements, several techniques such as *blind signatures* or *blind tokens* may be used. These would enable to grant permission to a collection of identities to perform an action, e.g. rate a review, without revealing which of them voted, or which voted for what. As previously, *single-use identities* may be used to provide anonymity; in this case, for raters.

Allowing only authorized rates, as previously explained, may help to prevent Sybil attacks (Challenge 4). Moreover, the cost of losing a reputable identity may reduce the attractiveness of starting a new identity just for the sake of reputation.

The use of the mentioned zk-SNARKs may also help to prevent some system abuses. For instance by enabling the use of cryptographic proofs that verify that the ratings come from reviews submitted to reputable journals, would prevent fake reviews and ratings.

Next, the previously mentioned technologies are explained.

**Single-use identities** New single-use identities may be used as a simple technique to support anonymous interactions (Challenge 1). However, supporting the authorization rules of the system (Challenge 3) and providing accountability (Challenge 2) for those identities are challenges that require consideration.

**Ring signatures** Ring signatures [130] are a cryptographic technique that allows to authorize a collection of identities to perform an action, while keeping the privacy of the specific identity that performed the action. They may be used to authorize rates to a group of identities, e.g. the authorized raters, without revealing who rated what or who rated. Thus, this technique may be used to support the authorization requirements of the system (Challenge 3), while providing some anonymity to the users (Challenge 1). Note that with this technique, the identities of who may have signed are known, so the combination with other anonymity measures could be of interest.

**Blind tokens** In the context of an election and using a cryptographic technique called blind signatures [131], it is possible to create ballots for authorized actors that preserve the anonymity of the vote (both hiding who casted a vote and

what each actor voted) but ensuring that only authorized voters participated. Note that, as with ring signatures, the identities of who may signed are known, and thus complementary anonymity measures could be used. This technique has been also used to anonymize a distributed reputation system [129]. Thus, it could be used to provide anonymity to reviewers and raters (Challenge 1) while supporting the authorization rules of the system, i.e. who may submit a review or a rating (Challenge 3).

**Collateral pattern** In order to secure the funds needed for a blockchain application to function, it is common that the application requests the participants to pay as collateral the assets they may lose. For instance, a betting smart contract will first ask all participants to pay their bets and afterwards distribute the prices. This paper calls this technique "collateral pattern", and proposes its use to provide accountability (Challenge 2) to the reviewers of the reputation system (see section 4.2.2). For each rating a reviewer may obtain, the reviewer must spend as much reputation as she may lose. This encourages the claiming of bad ratings, since not claiming them may result in a bigger loss.

**zk-SNARK** is a cryptographic procedure enabling to prove a statement without revealing anything else, i.e. apart from the evaluation if the statement is in fact true (zero-knowledge proof of knowledge) [132; 133]. The same authors also provide this property in a succinct and non-interactive fashion, i.e. using a relatively small proof and not requiring further communication between prover and verifier. In fact, the popular ZCash project uses this technology to build an anonymous cryptocurrency [134]. Proving statements in this privacy preserving manner is of great interest for several challenges of the proposed accountable anonymous review system. For instance, proving that a user controls a single-use identity (explained above) may allow the user to claim the reputation given to that identity (Challenge 1). Additionally, a reviewer may prove that she paid the reputation collateral (explained above) needed to submit a review without revealing her identity and without being able to use the same collateral for another review (Challenge 2). Finally, proving that the reputation comes from a review submitted to a collection of honest journals that do not allow abuses, may help to mitigate the abuses that fake reviews and ratings represents for the system (Challenge 5).



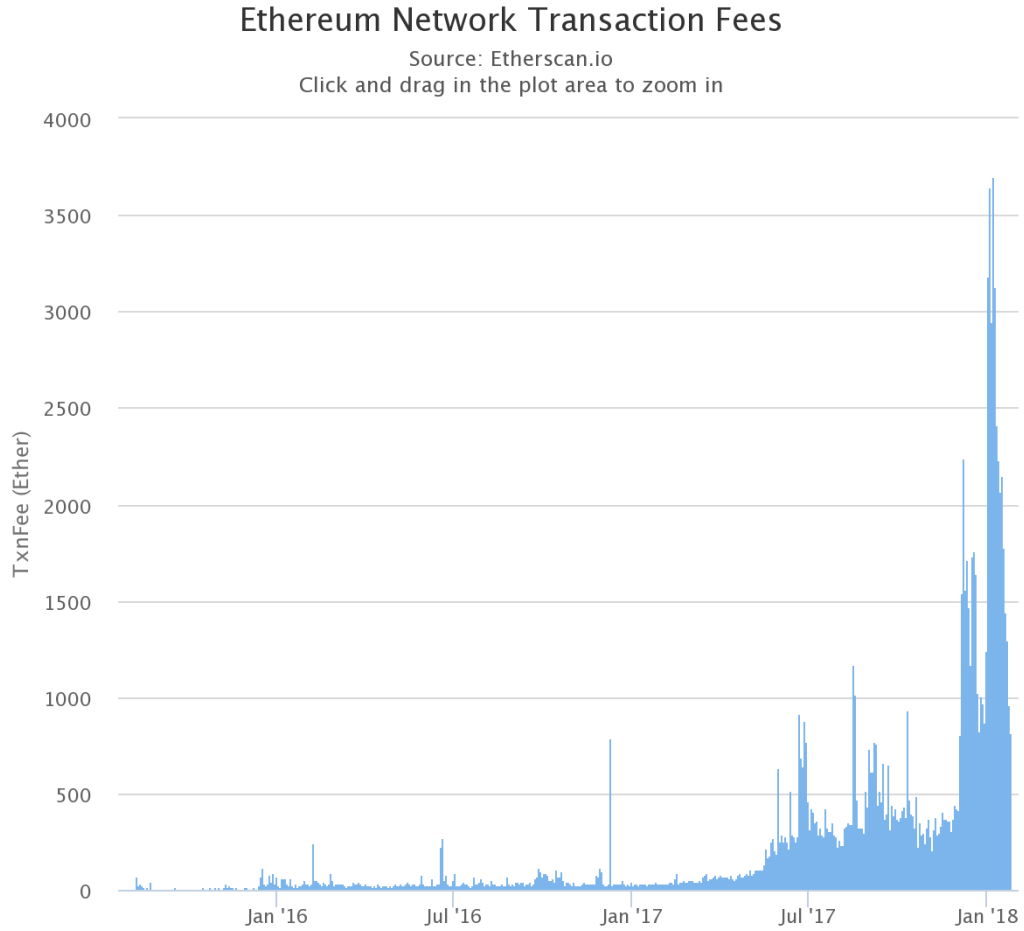


Figure 7.1: Ethereum transaction fees evolution.

#### 7.1.4 Other Problems

Another problem is that the proposed technologies still have little expansion and are little known by the average user.

For current researchers, a change of platform can be an inconvenience, since today's technologies such as *EasyChair*<sup>1</sup> can be difficult to replace. Proposing a change in the communication and revision systems can generate a great initial rejection, taking the project to a dead end, especially since today's methods to connect to Ethereum's blockchain are still complex. As a possible solution, Decentralized Science could be deployed as a demo in small and less known journals to introduce this

<sup>1</sup><https://easychair.org/>

system to the academic community. If one journal adopts this technology, it can be spread to other small journals, trying to extend this system as a substitute of the widely used ones. Furthermore, in order to improve the Ethereum's usability, a small framework to enhance user experience can be developed.

Another problem relating to the technology is that Ethereum's cryptocurrency fluctuates a lot, and lately, there has been a rise in all cryptocurrency prices compared to a year ago<sup>2</sup>. As shown in the figure 7.1 this price affects transactions, which may cause an increase of the prices to interact with the platform.

## 7.2 Monetary Impact

The monetary impact would be one of the most notable after the implementation of this system in today's scientific publishing systems. According to research data, the cost of publishing an article in a open access journal with a high impact index varies from 1,000 to 5,000 euros [15; 135], a cost that is often not feasible for researchers who want to advance in scientific research. In the other hand, the costs of reading papers published in non open access journals is usually unfair and expensive as explored in section 2.1.1.

The cost of publication and access to science through the proposed work would only vary depending on the price of the cryptocurrency used by Ethereum (see section 3.2.2). As seen in section 5.6, the maximum cost of a transaction in all the publication process would be a few euros. Bearing in mind that for a paper to be published at least 5 transactions must be carried out, it can be determined that the current price for publishing a paper is around 2 euros according to Ethereum Gas Station<sup>3</sup>, more than 500 times cheaper than the publishing system mentioned before.

## 7.3 Impact in the scientific community

Another important impact would be the reduction of time and the increase of quality in the process of peer review. If a critical mass of researchers use the proposed platform, the peer review process would be affected in two ways:

1. **Improvement in the review time:** Smart contracts allow establishing time limits for the review of an article, assuming a penalty to reviewers who do

---

<sup>2</sup>According to <http://coincap.io/>

<sup>3</sup>Price of a transaction in Ethereum <https://ethgasstation.info/>

not meet these deadlines (see section 4.1). If a Decentralized Journal has defined certain review times, and reviewers who are assigned accept the review proposal, it is likely that there will be an improvement in the delivery time of the reviews. Therefore the publication process will be faster regarding the delivery times of the reviews of today's publishing systems [7].

2. **Improvement in the review quality:** All reviews are rated by the community and directly affect the reputation of the reviewer. In this scenario it is likely that the quality of the reviews in the system will suffer an improvement and some problems regarding the peer review discussed in section 1 may be alleviated.

All interaction with the platform must be done through an Ethereum account and recorded in its blockchain. This would imply that through the address of a researcher you can obtain all the papers this researcher have published and reviewed. The entire scientific community could improve thanks to this platform because an Ethereum address could be used as an accreditation system. In addition, new researchers who want to start their career in academia can gain visibility thanks to the reputation system.

If the system were implemented successfully, the scientific community could begin to question the existence of the publishers and new forms of project funding could be found.

Finally, all the code developed for this platform is open source, meaning that it is totally free and accessible through the project github<sup>4</sup>. This implies several important things:

- Review submission, reputation system, journal delivery times and paper storage can be changed to other alternatives, allowing journals and conferences implement their own protocols and systems.
- The code can be improved and reviewed by other developers, opening submissions to find vulnerabilities and better implementations.

## 7.4 Platform legitimization

One of the main objectives is to legitimize the platform. offering the possibility to accredit the career of a researcher with the information contained in the system.

---

<sup>4</sup>Available at <https://decentralized.science>

No matter how much reputation a user has or how many papers she has published, if accrediting entities such as the aforementioned ANECA (see section 2.1) do not accept this system, these potential users will continue to use the current publication systems.

It would be possible to change this paradigm if any of these entities formed an alliance with the project. Maybe in the future, in order to prove that a person can teach in the university, she would only have to provide her Ethereum address.

## Chapter 8

# Conclusions and future work

*We don't want to change. Every change is a menace to stability.*

Aldous Huxley - Brave New World

### 8.1 Concluding remarks

This work proposes the opening and decentralization of three of the peer review and publication functions: 1) the peer review process communication, 2) the reputation of reviewers, and 3) the distribution of papers and peer reviews. It offers a first approach to the platform's implementation with a functional prototype.

Distributed technologies such as Blockchain and IPFS may finally realize the promise of Open Access, while enabling new models of science dissemination.

Opening and decentralizing the infrastructure enhances the transparency and accountability of the system, and may provide a new arena to foster innovation.

The transparency provided by opening the peer review process allows the construction of a reputation system of reviewers, but also raises concerns about privacy and fairness. Besides, the introduction of a new public metric (reviewers' reputation) may also affect researcher careers, adding pressure to the already straining processes for academic survival [136].

Blockchain technologies can be used to replicate the privacy settings currently used in peer review processes. However, Blockchain can also be used to introduce a new review model that supports the accountability of peer reviewing while keeping the anonymity of blind and double blind reviews to improve fairness. The implications of such accountable, open and anonymous review models are still to be revealed.

Additionally, the system's infrastructure relies on new technologies with their own challenges. Blockchain technologies face scalability, transaction costs, inclusiveness and usability problems that remain open and under discussion. On the other hand, distributed file systems such as IPFS may be more resilient, but they still need somebody in charge of preserving and providing the data, since without that responsible actor, these systems may result in unpredictable loss of content.

Other open issues that may be explored in future work are the exploration of different copyright regimes, the challenging of traditional journal-centered metrics to rate publication quality, different reputation algorithms, different levels of openness, and the exploration of decentralized autonomous journals.

Despite the existing challenges, decentralizing the processes that Science relies on could open up a whole new playing field, with implications we cannot possibly foresee now. Will its benefits outweigh its risks?

## 8.2 Future Work

One of the first problems to attack would be the anonymity in the peer reviews, because no matter how utopian it may seem, a completely public system in which there is no anonymity of the reviewers and the authors is quite difficult to implement. This problem could be mitigated with some of the solutions proposed in the section 7.1.2. Nevertheless, Decentralized Science's source code can be migrated to other blockchain technologies. Thus, if in the future, there a similar technology to Ethereum, where anonymity protocols are already implemented, a version of this platform could be easily implemented.

As extensions to this work, the following ideas are proposed:

- The incorporation of a system to forward papers already accepted, trying to expand the research they propose, including the possibility of being able to follow the research lines of other authors, completing their papers to build a collaborative scientific community, based on mutual support.
- A standard citation methodology in which both papers and authors are Ethereum's addresses replacing identifiers like ISBN or DOI. An author address could give information about the author's work, offering the possibility to view the papers she sent, the ones she reviewed and collaborations made with other authors. A paper's address could give information about the impact factor it has, showing all other papers that cite it.

- The automation of the assignment of reviewers for a newly submitted article, so that the formal role of the editor disappears, since the choice will depend entirely on the smart contract of the journal to which it corresponds. This choice could be based on the trust that the community has in the reviewers' reputation network, being able to eliminate one of the intermediaries from the system.
- Finally, the intermediary of the journals could be eliminated. If the assignment of reviewers and publication of the articles is automatic and is in the blockchain, the existence of the journal could be questioned by the scientific community. With the support of a fully functional and widely used reputation system for reviewers, an ideal way of scientific publishing could be a large library in which authors submit their paper, reviewers are chosen based on their reputation, and papers are automatically published.

To conclude this work, I propose a future PhD with new development lines with higher impact in the scientific community. This proposal would include: an exhaustive exploration of all blockchain technologies, interviews with real users to validate the platform, social simulation using multi-agent systems to analyze the platform's behavior and reaching potential users to deploy demos in real environments.

Within a few years, scientific publication methods may be based on projects such as the one presented in this work, changing the paradigm that has been imposed for so many years in the scientific research process.

### 8.3 Conferences and Papers

This project has presented the following papers:

- Tenorio-Fornés, A., Jacynycz, V., Llop, D., Sánchez-Ruiz, A.A., Hassan, S. **“A Decentralized Publication System for Open Science using Blockchain and IPFS”**. *PEERE International Conference on Peer Review Proceedings*. Rome (2018).
- Tenorio-Fornés, A., Jacynycz, V., Llop, D., Sánchez-Ruiz, A.A., Hassan, S. **“Towards a Decentralized Process for Scientific Publication and Peer Review using Blockchain and IPFS”**. *HICSS Hawaii International Conference on System Sciences (CORE-A)*. Hawaii (2019).

Decentralized Science project has been presented at the following community conferences:

- **EthCC2018** - Ethereum community conference. *Paris, France* (March 2018). <https://ethcc.io/>
- **SPONBC2018** - Scientific Publishing on the Blockchain. *Vienna, Austria* (May 2018). <https://www.blockchainforscience.com/2018/02/09/sponbc2018/>

## 8.4 Acknowledgements

This project is financed by the Complutense University of Madrid (CT2/17-CT3/17) and partially supported by the project P2P Models (<https://p2pmodels.eu>) funded by the European Research Council ERC-2017-STG (grant no.: 759207), the Spanish Ministry of Economy, Industry and Competitiveness (TIN2014-55006-R, TIN2017-87330-R) and GRASIA UCM (Group 921330).



## 8.5 Conclusiones finales

Este trabajo propone la descentralización de tres de las funciones del proceso de revisión por pares: 1) la comunicación del proceso de revisión por pares, 2) la reputación de los revisores, y 3) la distribución de los artículos y las revisiones de estos. Ofrece un primer acercamiento a la implementación de la plataforma con un prototipo funcional.

Las tecnologías distribuidas como Blockchain e IPFS pueden cumplir las promesas del *Open Access* a la vez que abren nuevos modelos para la distribución de ciencia.

La descentralización de la estructura mejora la transparencia y la honestidad del sistema, y puede proporcionar nuevos escenarios para fomentar la innovación. También se ha de tener en cuenta que el sistema y los prototipos propuestos no se basan en el uso de criptomonedas, ya que se centran en un enfoque sin fines lucrativos, lejos de los enfoques comerciales impulsados por las startups, cada vez más comunes dentro en el espacio blockchain.

La transparencia proporcionada al abrir el proceso de revisión por pares permite la construcción de un sistema de reputación de revisores, pero también plantea problemas sobre la privacidad y la equidad. Además, la introducción de una nueva métrica pública (la reputación de los revisores) también puede afectar las carreras de los investigadores, lo que puede suponer un aumento en la presión a los procesos ya de por sí agotadores para la supervivencia académica [136].

Las tecnologías de Blockchain se pueden usar para replicar la configuración de privacidad utilizada actualmente en los procesos de revisión por pares. Sin embargo, Blockchain también se puede utilizar para presentar un nuevo modelo de revisión que respalde la responsabilidad de la revisión por pares y al mismo tiempo mantener el anonimato de las revisiones a ciegas y doble ciego para mejorar la equidad. Las implicaciones de estos posibles modelos de revisión responsables, abiertas y anónimas aún están por revelarse.

Además, la infraestructura del sistema se basa en nuevas tecnologías con sus propios desafíos. Las tecnologías de Blockchain se enfrentan a problemas de escalabilidad, costos de transacción, inclusión y usabilidad. Por otro lado, los sistemas de archivos distribuidos como IPFS pueden ser más resistentes, pero aún necesitan a alguien a cargo de preservar y proporcionar los datos, ya que sin esa persona responsable, puede provocar la pérdida imprevista de contenido almacenado por las plataformas que utilicen dicho sistema de archivos.

Otros temas abiertos que pueden explorarse en el trabajo futuro son la exploración

de diferentes modelos de derechos de autor, el desafío de las métricas tradicionales centradas en revistas para calificar la calidad de la publicación, diferentes algoritmos de reputación, diferentes niveles de apertura y la exploración de revistas autónomas descentralizadas, capaces de operar de manera automática sin interacción de los usuarios.

A pesar de los desafíos existentes, la descentralización de los procesos en los que se basa la ciencia podría abrir un nuevo campo de exploración, con implicaciones que posiblemente no podamos prever ahora. ¿Podrán los beneficios superar a los riesgos?

## 8.6 Trabajo futuro

Uno de los primeros problemas a atacar sería el anonimato en las revisiones por pares, porque no importa lo utópico que parezca, un sistema completamente público en el que no hay anonimato de los revisores y los autores es difícil de implementar. Este problema podría mitigarse con algunas de las soluciones propuestas en la sección 7.1.2. Sin embargo, el código fuente de este trabajo se puede migrar a otras tecnologías de blockchain. Por lo tanto, si en el futuro, existe una tecnología similar a Ethereum, donde los protocolos de anonimato ya están implementados, una versión de esta plataforma podría implementarse fácilmente.

Como posibles extensiones de este proyecto se proponen las siguientes ideas:

- La incorporación de un sistema para enviar documentos ya aceptados, tratando de ampliar la investigación que proponen, incluyendo la posibilidad de seguir las líneas de investigación de otros autores, completando sus trabajos para construir una comunidad científica colaborativa, basada en el apoyo mutuo.
- Una metodología de citación estándar en la que tanto documentos como autores son direcciones de Ethereum que reemplazan identificadores como ISBN o DOI. Una dirección de autor podría proporcionar información sobre el trabajo del autor, ofreciendo la posibilidad de ver los trabajos que envió, los que revisó y las colaboraciones realizadas con otros autores. La dirección de un artículo podría proporcionar información sobre el factor de impacto que tiene, mostrando todos los demás artículos que lo citan.
- La automatización de la asignación de revisores para un artículo recién enviado, de modo que el papel formal del editor desaparezca, ya que la elección dependerá completamente del contrato inteligente de la revista a la que corresponda.

Esta elección podría basarse en la confianza que la comunidad tiene en la red de reputación de los revisores, pudiendo eliminar a uno de los intermediarios del sistema.

- Finalmente, el intermediario de las revistas científicas podría eliminarse. Si la asignación de revisores y la publicación de los artículos es automática y está en el blockchain, la comunidad científica cuestionaría la existencia de estas revistas. Con el apoyo de un sistema de reputación totalmente funcional y ampliamente utilizado para los revisores, una forma ideal de publicación científica podría ser una gran biblioteca en la que los autores presenten su trabajo, los revisores se elijan al azar según su reputación y los trabajos se publiquen automáticamente.

Para concluir este trabajo, propongo un futuro doctorado con nuevas líneas de desarrollo con mayor impacto en la comunidad. Esta propuesta incluiría: una exploración exhaustiva de todas las tecnologías de blockchain, entrevistas con usuarios reales para validar la plataforma, simulación social usando sistemas multi-agente para analizar el comportamiento de la plataforma y contactar con usuarios potenciales para implementar demos en entornos reales.

Puede de dentro de unos años los métodos de publicación científica se basen en proyectos como el que presenta este trabajo, cambiando el paradigma que lleva tantos años impuesto en el proceso de investigación científica.

## 8.7 Conferencias y artículos académicos

Este proyecto ha presentado los siguientes artículos académicos:

- Tenorio-Fornés, A., Jacynycz, V., Llop, D., Sánchez-Ruiz, A.A., Hassan, S. **“A Decentralized Publication System for Open Science using Blockchain and IPFS”**. *PEERE International Conference on Peer Review Proceedings*. Rome (2018).
- Tenorio-Fornés, A., Jacynycz, V., Llop, D., Sánchez-Ruiz, A.A., Hassan, S. **“Towards a Decentralized Process for Scientific Publication and Peer Review using Blockchain and IPFS”**. *HICSS Hawaii International Conference on System Sciences (CORE-A)*. Hawaii (2019).

El proyecto de Decentralized Science ha sido presentado en las siguientes conferencias de la comunidad:

- **EthCC2018** - Ethereum community conference. *París, Francia* (Marzo 2018).  
<https://ethcc.io/>
- **SPONBC2018** - Scientific Publishing on the Blockchain. *Viena, Austria* (Mayo 2018).  
<https://www.blockchainforscience.com/2018/02/09/sponbc2018/>

## 8.8 Agradecimientos

Este proyecto ha sido financiado por la Universidad Complutense de Madrid(CT2/17-CT3/17) y apoyado parcialmente por el proyecto P2P Models (<https://p2pmodels.eu>) financiado por el *European Research Council* ERC-2017-STG (grant no.: 759207), el ministerio de economía y competitividad de España (TIN2014-55006-R, TIN2017-87330-R) y GRASIA UCM (Grupo 921330).





# Bibliography

*You act, and you know why you act, but you don't know  
why you know that you know what you do?*

The name of the rose - Umberto eco

- [1] G. E, “The history and meaning of the journal impact factor,” *JAMA*, vol. 295, no. 1, pp. 90–93, 2006. [Online]. Available: [+http://dx.doi.org/10.1001/jama.295.1.90](http://dx.doi.org/10.1001/jama.295.1.90)
- [2] E. Garfield, “The evolution of the science citation index,” *International microbiology*, vol. 10, no. 1, p. 65, 2007.
- [3] B. S. Frey, “Publishing as prostitution? – choosing between one’s own ideas and academic success,” *Public Choice*, vol. 116, no. 1, pp. 205–223, Jul 2003. [Online]. Available: <https://doi.org/10.1023/A:1024208701874>
- [4] D. J. Ortinau, “Writing and publishing important scientific articles: A reviewer’s perspective,” *Journal of Business Research*, vol. 64, no. 2, pp. 150–156, 2011.
- [5] R. Spier, “The history of the peer-review process,” *TRENDS in Biotechnology*, vol. 20, no. 8, pp. 357–358, 2002.
- [6] C. Wenneras and A. Wold, “Nepotism and sexism in peer-review,” *Women, science and technology: A reader in feminist science studies*, pp. 46–52, 2001.
- [7] J. Huisman and J. Smits, “Duration and quality of the peer review process: the author’s perspective,” *Scientometrics*, pp. 1–18, 2017.
- [8] S. Goldbeck-Wood, “Evidence on peer review—scientific quality control or smokescreen?” *BMJ: British Medical Journal*, vol. 318, no. 7175, p. 44, 1999.

- [9] G. Eysenbach, "Citation advantage of open access articles," *PLoS biology*, vol. 4, no. 5, p. e157, 2006.
- [10] R. Walker and P. Rocha da Silva, "Emerging trends in peer review—a survey," *Frontiers in Neuroscience*, vol. 9, May 2015.
- [11] B. Whitworth and R. Friedman, "Reinventing academic publishing online. part i: Rigor, relevance and practice," *First Monday*, vol. 14, no. 8, 2009. [Online]. Available: <http://firstmonday.org/ojs/index.php/fm/article/view/2609>
- [12] C. T. Bergstrom and T. C. Bergstrom, "The costs and benefits of library site licenses to academic journals," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 3, pp. 897–902, 2004.
- [13] J. A. Evans and J. Reimer, "Open access and global participation in science," *Science*, vol. 323, no. 5917, pp. 1025–1025, 2009.
- [14] V. Larivière, S. Haustein, and P. Mongeon, "The oligopoly of academic publishers in the digital era," *PloS one*, vol. 10, no. 6, p. e0127502, 2015.
- [15] R. Van Noorden *et al.*, "The true cost of science publishing," *Nature*, vol. 495, no. 7442, pp. 426–429, 2013.
- [16] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, 2000.
- [17] M. L. Callahan and J. Tercier, "The relationship of previous training and experience of journal peer reviewers to subsequent review quality," *PLoS medicine*, vol. 4, no. 1, p. e40, 2007.
- [18] M. Ware, "Peer review: benefits, perceptions and alternatives," *Publishing Research Consortium*, vol. 4, pp. 1–20, 2008.
- [19] E. Ford, "Defining and characterizing open peer review: A review of the literature," *Journal of Scholarly Publishing*, vol. 44, no. 4, pp. 311–326, 2013.
- [20] P. Frishauf, "Reputation systems: a new vision for publishing and peer review," *Journal of Participatory Medicine*, 2009.
- [21] F. Bacon, *Novum organum*. Clarendon press, 1878.
- [22] D. A. Kronick, "A history of scientific and technical periodicals: the origins and development of the scientific and technological press, 1665-1790," 1976.



- [23] J. C. Burnham, "The evolution of editorial peer review," *Jama*, vol. 263, no. 10, pp. 1323–1329, 1990.
- [24] M. Szklo, "Quality of scientific articles," *Revista de Saúde Pública*, vol. 40, no. SPE, pp. 30–35, 2006.
- [25] S. Redner, "How popular is your paper? an empirical study of the citation distribution," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 4, no. 2, pp. 131–134, 1998.
- [26] L. Bornmann and H.-D. Daniel, "What do we know about the h index?" *Journal of the Association for Information Science and Technology*, vol. 58, no. 9, pp. 1381–1385, 2007.
- [27] L. Bornmann, R. Mutz, and H.-D. Daniel, "Are there better indices for evaluation purposes than the h index? a comparison of nine different variants of the h index using data from biomedicine," *Journal of the Association for Information Science and Technology*, vol. 59, no. 5, pp. 830–837, 2008.
- [28] J. E. Hirsch, "Does the h index have predictive power?" *Proceedings of the National Academy of Sciences*, vol. 104, no. 49, pp. 19 193–19 198, 2007.
- [29] H. R. Rothstein and S. Hopewell, "Grey literature." 2009.
- [30] C. W. Bailey Jr, "What is open access?" 2007.
- [31] P. Binfield, "Open access megajournals: have they changed everything?" 2013.
- [32] R. Wellen, "Open access, megajournals, and moocs: on the political economy of academic unbundling," *SAGE Open*, vol. 3, no. 4, p. 2158244013507271, 2013.
- [33] B.-C. Björk, "Have the "mega-journals" reached the limits to growth?" *PeerJ*, vol. 3, p. e981, 2015.
- [34] S. Anderton and L. Harvey, "Continuous publication: ready, set, cite!" 2013.
- [35] R. Hayman, "View the continuous publication model as a satisfactory alternative for open access lis journals. evid based libr inf pract. 2014; 9."
- [36] S. Harnad, "Electronic preprints and postprints," in *Encyclopedia of library and information science*. Marcel Dekker, 2003, vol. 67, no. 4.

- [37] C. Brown, “The e-volution of preprints in the scholarly communication of physicists and astronomers,” *Journal of the Association for Information Science and Technology*, vol. 52, no. 3, pp. 187–200, 2001.
- [38] X. Shuai, A. Pepe, and J. Bollen, “How the scientific community reacts to newly submitted preprints: Article downloads, twitter mentions, and citations,” *PloS one*, vol. 7, no. 11, p. e47523, 2012.
- [39] K. Sugiyama and M.-Y. Kan, “Scholarly paper recommendation via user’s recent research interests,” in *Proceedings of the 10th annual joint conference on Digital libraries*. ACM, 2010, pp. 29–38.
- [40] A. Josang and R. Ismail, “The beta reputation system,” in *Proceedings of the 15th bled electronic commerce conference*, vol. 5, 2002, pp. 2502–2511.
- [41] P. Resnick and R. Zeckhauser, “Trust among strangers in internet transactions: Empirical analysis of ebay’s reputation system,” in *The Economics of the Internet and E-commerce*. Emerald Group Publishing Limited, 2002, pp. 127–157.
- [42] R. Zhou and K. Hwang, “Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing,” *IEEE Transactions on parallel and distributed systems*, vol. 18, no. 4, 2007.
- [43] F. Dotzer, L. Fischer, and P. Magiera, “Vars: A vehicle ad-hoc network reputation system,” in *World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a*. IEEE, 2005, pp. 454–456.
- [44] G. Moore and L. G. Meredith, “Reputation system for web services,” Dec. 16 2008, uS Patent 7,467,206.
- [45] B. T. Adler and L. De Alfaro, “A content-driven reputation system for the wikipedia,” in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 261–270.
- [46] M. Sharples and J. Domingue, “The blockchain and kudos: A distributed system for educational record, reputation and reward,” in *European Conference on Technology Enhanced Learning*. Springer, 2016, pp. 490–496.
- [47] K. Hoffman, D. Zage, and C. Nita-Rotaru, “A survey of attack and defense

- techniques for reputation systems,” *ACM Computing Surveys (CSUR)*, vol. 42, no. 1, p. 1, 2009.
- [48] A. Whitby, A. Jøsang, and J. Indulska, “Filtering out unfair ratings in bayesian reputation systems,” in *Proc. 7th Int. Workshop on Trust in Agent Societies*, vol. 6, 2004, pp. 106–117.
- [49] M. Kinateder and K. Rothermel, “Architecture and algorithms for a distributed reputation system,” in *International Conference on Trust Management*. Springer, 2003, pp. 1–16.
- [50] P. Baran, “On distributed communications networks,” *IEEE transactions on Communications Systems*, vol. 12, no. 1, pp. 1–9, 1964.
- [51] R. Schollmeier, “A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications,” in *Peer-to-Peer Computing, 2001. Proceedings. First International Conference on*. IEEE, 2001, pp. 101–102.
- [52] D. Chaum, “An introduction to ecash,” *DigiCash*, <http://www.digicash.com>, 1995.
- [53] W. Dai, “B-money proposal,” 1998.
- [54] A. Back *et al.*, “Hashcash-a denial of service counter-measure,” 2002.
- [55] N. Szabo, “Bit gold,” *Website/Blog*, 2008.
- [56] H. Finney, “Rpow: Reusable proofs of work,” *CodeCon 2005*, 2005.
- [57] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.
- [58] U. Chohan, “The double spending problem and cryptocurrencies. ssrn,” 2017.
- [59] S. Barber, X. Boyen, E. Shi, and E. Uzun, “Bitter to better—how to make bitcoin a better currency,” in *International Conference on Financial Cryptography and Data Security*. Springer, 2012, pp. 399–414.
- [60] A. M. Antonopoulos, *Mastering Bitcoin: unlocking digital cryptocurrencies*. " O'Reilly Media, Inc.", 2014.
- [61] T. Moore and N. Christin, “Beware the middleman: Empirical analysis of bitcoin-exchange risk,” in *International Conference on Financial Cryptography and Data Security*. Springer, 2013, pp. 25–33.

- [62] B. Benshoof, A. Rosen, A. G. Bourgeois, and R. W. Harrison, “Distributed decentralized domain name service,” in *Parallel and Distributed Processing Symposium Workshops, 2016 IEEE International*. Chicago, IL, USA: IEEE, 2016, pp. 1279–1287.
- [63] M. Freitas, “twister-a p2p microblogging platform,” *arXiv preprint arXiv:1312.7152*, 2013.
- [64] V. Buterin, “Ethereum: A next-generation smart contract and decentralized application platform,” 2014.
- [65] P. J. Piasecki, “Gaming self-contained provably fair smart contract casinos,” *Ledger*, vol. 1, pp. 99–110, 2016.
- [66] P. McCorry, S. F. Shahandashti, and F. Hao, “A smart contract for boardroom voting with maximum voter privacy,” in *International Conference on Financial Cryptography and Data Security*. Springer, 2017, pp. 357–375.
- [67] V. Jacynycz, A. Calvo, S. Hassan, and A. A. Sánchez-Ruiz, “Betfunding: A distributed bounty-based crowdfunding platform over ethereum,” in *Distributed Computing and Artificial Intelligence, 13th International Conference*. Springer, 2016, pp. 403–411.
- [68] J. Peterson and J. Krug, “Augur: a decentralized, open-source platform for prediction markets,” *arXiv preprint arXiv:1501.01042*, 2015.
- [69] J. Bonneau, “Ethiks: Using ethereum to audit a coniks key transparency log,” in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 95–105.
- [70] G. Wood, “Ethereum: A secure decentralised generalised transaction ledger,” *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, 2014.
- [71] K. Delmolino, M. Arnett, A. Kosba, A. Miller, and E. Shi, “Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab,” in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 79–94.
- [72] O. H. Azar, “The academic review process: How can we make it more efficient?” *The American Economist*, vol. 50, no. 1, pp. 37–50, 2006.

- [73] (2002) Easychair conference system - easychair ltd. [Online]. Available: <http://www.easychair.org>
- [74] (2004) Openconf event management system - zakon group llc. [Online]. Available: <https://www.openconf.com/>
- [75] P. Cristo and R. Martinho, "Aems: Towards an advanced event management system," in *ENTERprise Information Systems*, M. M. Cruz-Cunha, J. Varajão, P. Powell, and R. Martinho, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 82–90.
- [76] B. Lev, "From rmts to ees to evise," *Omega*, vol. 61, no. Complete, p. 1, 2016.
- [77] (2016) Evise web-based editorial system - elsevier. [Online]. Available: <https://www.elsevier.com/editors/evise>
- [78] Myspringer - springer nature switzerland ag. [Online]. Available: <https://www.springer.com/gp/myspringer>
- [79] J. Willinsky, "Open journal systems: An example of open source software for journal management and publishing," *Library hi tech*, vol. 23, no. 4, pp. 504–519, 2005.
- [80] E. Rajpert-De Meyts, S. Losito, and D. T. Carrell, "Rewarding peer-review work: the publons initiative," *Andrology*, vol. 4, no. 6, pp. 985–986, 2016.
- [81] J. Sabater-Mir, F. Grimaldo, and M. Paolucci, "Peer review as a reputation system."
- [82] S. Bartling and B. Fecher, "Blockchain for science and knowledge creation. zenodo," *Publisher Full Text*, 2016.
- [83] J. P. Tennant, J. M. Dugan, D. Graziotin, D. C. Jacques, F. Waldner, D. Mitichen, Y. Elkhatib, L. B. Collister, C. K. Pikas, T. Crick *et al.*, "A multi-disciplinary perspective on emergent and future innovations in peer review," *F1000Research*, vol. 6, 2017.
- [84] A. C. Kade Morton, "Aletheia: blockchain for scientific knowledge with a community management framework," 2017. [Online]. Available: <https://github.com/aletheia-foundation/aletheia-whitepaper/>

- 
- [85] V. Dhillon, "From bench to bedside: Enabling reproducible commercial science via blockchain," *Bitcoin Magazine*, 2016.
- [86] M. Fuster Morell, "Governance of online creation communities: Provision of infrastructure for the building of digital commons," Ph.D. dissertation, European University Institute, 2010.
- [87] A. F. Osborn, "Applied imagination." 1953.
- [88] R. I. Sutton and A. Hargadon, "Brainstorming groups in context: Effectiveness in a product design firm," *Administrative Science Quarterly*, pp. 685–718, 1996.
- [89] B. Mullen, C. Johnson, and E. Salas, "Productivity loss in brainstorming groups: A meta-analytic integration," *Basic and applied social psychology*, vol. 12, no. 1, pp. 3–23, 1991.
- [90] J. Pokorná, L. Pilar, T. Balcarová, and I. Sergeeva, "Value proposition canvas: Identification of pains, gains and customer jobs at farmers' markets," *AGRIS on-line Papers in Economics and Informatics*, vol. 7, no. 4, p. 123, 2015.
- [91] L. O. Meertens, M. E. Iacob, L. J. M. Nieuwenhuis, M. J. van Sinderen, H. Jonkers, and D. Quartel, "Mapping the business model canvas to archimate," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, ser. SAC '12. New York, NY, USA: ACM, 2012, pp. 1694–1701. [Online]. Available: <http://doi.acm.org/10.1145/2245276.2232049>
- [92] A. Osterwalder, Y. Pigneur, G. Bernarda, and A. Smith, *Value proposition design: How to create products and services customers want*. John Wiley & Sons, 2014.
- [93] B. Boehm and R. Turner, "Management challenges to implementing agile processes in traditional development organizations," *IEEE software*, vol. 22, no. 5, pp. 30–39, 2005.
- [94] J. A. Livermore, "Factors that significantly impact the implementation of an agile software development methodology," *Journal of software*, vol. 3, no. 4, pp. 31–36, 2008.
- [95] L. Lindstrom and R. Jeffries, "Extreme programming and agile software development methodologies," *Information systems management*, vol. 21, no. 3, pp. 41–52, 2004.

- [96] W. Theunissen, A. Boake, and D. G. Kourie, "In search of the sweet spot: agile open collaborative corporate software development," in *Proceedings of the 2005 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*. South African Institute for Computer Scientists and Information Technologists, 2005, pp. 268–277.
- [97] L. Rising and N. S. Janoff, "The scrum software development process for small teams," *IEEE software*, vol. 17, no. 4, pp. 26–32, 2000.
- [98] M. O. Ahmad, J. Markkula, and M. Oivo, "Kanban in software development: A systematic literature review," in *Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on*. IEEE, 2013, pp. 9–16.
- [99] J. Benet, "IPFS-content addressed, versioned, P2p file system," *arXiv preprint arXiv:1407.3561*, 2014. [Online]. Available: <https://arxiv.org/abs/1407.3561>
- [100] —, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.
- [101] A. Kaluszka, "Distributed hash tables," 2010.
- [102] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," *IEEE Journal on selected areas in communications*, vol. 22, no. 1, pp. 41–53, 2004.
- [103] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4, pp. 149–160, 2001.
- [104] I. Gupta, K. Birman, P. Linga, A. Demers, and R. Van Renesse, "Kelips: Building an efficient and stable p2p dht through increased memory and background overhead," *Peer-to-Peer Systems II*, pp. 160–169, 2003.
- [105] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *International Workshop on Peer-to-Peer Systems*. Springer, 2002, pp. 53–65.
- [106] J. Li, J. Stribling, T. M. Gil, R. Morris, and M. F. Kaashoek, "Comparing the

- performance of distributed hash tables under churn.” in *Iptps*, vol. 4. Springer, 2004, pp. 87–99.
- [107] B. Cohen, “Incentives build robustness in bittorrent,” in *Workshop on Economics of Peer-to-Peer systems*, vol. 6, 2003, pp. 68–72.
- [108] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, “The bittorrent p2p file-sharing system: Measurements and analysis,” in *IPTPS*, vol. 5. Springer, 2005, pp. 205–216.
- [109] D. Levin, K. LaCurts, N. Spring, and B. Bhattacharjee, “Bittorrent is an auction: analyzing and improving bittorrent’s incentives,” in *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4. ACM, 2008, pp. 243–254.
- [110] L. Torvalds and J. Hamano, “Git: Fast version control system,” URL <http://git-scm.com>, 2010.
- [111] D. Spinellis, “Version control systems,” *IEEE Software*, vol. 22, no. 5, pp. 108–109, 2005.
- [112] D. D. F. Mazières, “Self-certifying file system,” Ph.D. dissertation, Massachusetts Institute of Technology, 2000.
- [113] E. Hildenbrandt, M. Saxena, X. Zhu, N. Rodrigues, P. Daian, D. Guth, and G. Rosu, “Kevm: A complete semantics of the ethereum virtual machine,” Tech. Rep., 2017.
- [114] H.-D. Daniel and W. E. Russey, *Guardians of science: Fairness and reliability of peer review*. Wiley Online Library, 1993.
- [115] J. R. Cole, “Fair science: Women in the scientific community,” 1979.
- [116] D. J. Solomon and B.-C. Björk, “A study of open access journals using article processing charges,” *Journal of the Association for Information Science and Technology*, vol. 63, no. 8, pp. 1485–1495, 2012.
- [117] B.-C. Björk, M. Laakso, P. Welling, and P. Paetau, “Anatomy of green open access,” *Journal of the Association for Information Science and Technology*, vol. 65, no. 2, pp. 237–250, 2014.
- [118] E. Rajpert-De Meyts, S. Losito, and D. T. Carrell, “Rewarding peer-review work: the Publons initiative,” *Andrology*, vol. 4, no. 6, pp. 985–986, Nov. 2016.



- [119] A. Tenorio-Fornés, V. Jacynycz, D. Llop, A. A. Sánchez-Ruiz, and S. Hassan, "A decentralized publication system for open science using blockchain and ipfs," in *PEERE International Conference on Peer Review*, 2018.
- [120] B. Whitworth and R. Friedman, "Reinventing academic publishing online. part ii: A socio-technical vision," *First Monday*, vol. 14, no. 9, 2009. [Online]. Available: <http://firstmonday.org/ojs/index.php/fm/article/view/2642>
- [121] Z. Jan, A. Third, L.-D. Ibanez, M. Bachler, E. Simperl, and J. Domingue, "Sciencemiles: Digital currency for researchers," in *Companion Proceedings of the The Web Conference 2018*, ser. WWW '18. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2018, pp. 1183–1186. [Online]. Available: <https://doi.org/10.1145/3184558.3191556>
- [122] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer *et al.*, "On scaling decentralized blockchains," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 106–125.
- [123] E. S. Gardner, "Exponential smoothing: The state of the art," *Journal of forecasting*, vol. 4, no. 1, pp. 1–28, 1985.
- [124] C. Dellarocas, "Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior," in *Proceedings of the 2nd ACM conference on Electronic commerce*. ACM, 2000, pp. 150–157.
- [125] C. J. Lee, C. R. Sugimoto, G. Zhang, and B. Cronin, "Bias in peer review," *Journal of the Association for Information Science and Technology*, vol. 64, no. 1, pp. 2–17, 2013.
- [126] A. E. Budden, T. Tregenza, L. W. Aarssen, J. Koricheva, R. Leimu, and C. J. Lortie, "Double-blind review favours increased representation of female authors," *Trends in ecology & evolution*, vol. 23, no. 1, pp. 4–6, 2008.
- [127] S. Van Rooyen, F. Godlee, S. Evans, N. Black, and R. Smith, "Effect of open peer review on quality of reviews and on reviewers' recommendations: a randomised trial," *Bmj*, vol. 318, no. 7175, pp. 23–27, 1999.
- [128] T. Groves and K. Khan, "Is open peer review the fairest system," *BMJ (Clinical research ed.)*, vol. 341, pp. 1082–1083, 2010.

- [129] A. Schaub, R. Bazin, O. Hasan, and L. Brunie, “A trustless privacy-preserving reputation system,” in *IFIP International Information Security and Privacy Conference*. Springer, 2016, pp. 398–411.
- [130] R. L. Rivest, A. Shamir, and Y. Tauman, “How to leak a secret,” in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2001, pp. 552–565.
- [131] D. Chaum, “Blind signatures for untraceable payments,” in *Advances in cryptography*. Springer, 1983, pp. 199–203.
- [132] M. Blum, P. Feldman, and S. Micali, “Non-interactive zero-knowledge and its applications,” in *Proceedings of the twentieth annual ACM symposium on Theory of computing*. ACM, 1988, pp. 103–112.
- [133] N. Bitansky, A. Chiesa, Y. Ishai, O. Paneth, and R. Ostrovsky, “Succinct non-interactive arguments via linear interactive proofs,” in *Theory of Cryptography*. Springer, 2013, pp. 315–333.
- [134] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, “Zerocash: Decentralized anonymous payments from bitcoin,” in *Security and Privacy (SP), 2014 IEEE Symposium on*. IEEE, 2014, pp. 459–474.
- [135] R. Russel, “The business of academic publishing: A strategic analysis of the academic journal publishing industry and its impact on the future of scholarly publishing,” *Electron J Acad Spec Librarianship*. <http://southernlibrarianship.icaap.org/content>, 2008.
- [136] M. De Rond and A. N. Miller, “Publish or perish: bane or boon of academic life?” *Journal of Management Inquiry*, vol. 14, no. 4, pp. 321–329, 2005.

*Until they become conscious they will never rebel,  
and until after they have rebelled they cannot become conscious.*

*1984*  
*George Orwell*

*These people don't see that if you encourage totalitarian methods,  
the time may come when they will be used against you instead of for you.*

*Animal Farm*  
*George Orwell*

